MAKE | BUILD | HACK | CREATE

132 PAGES EVERY MONTH

HackSpace

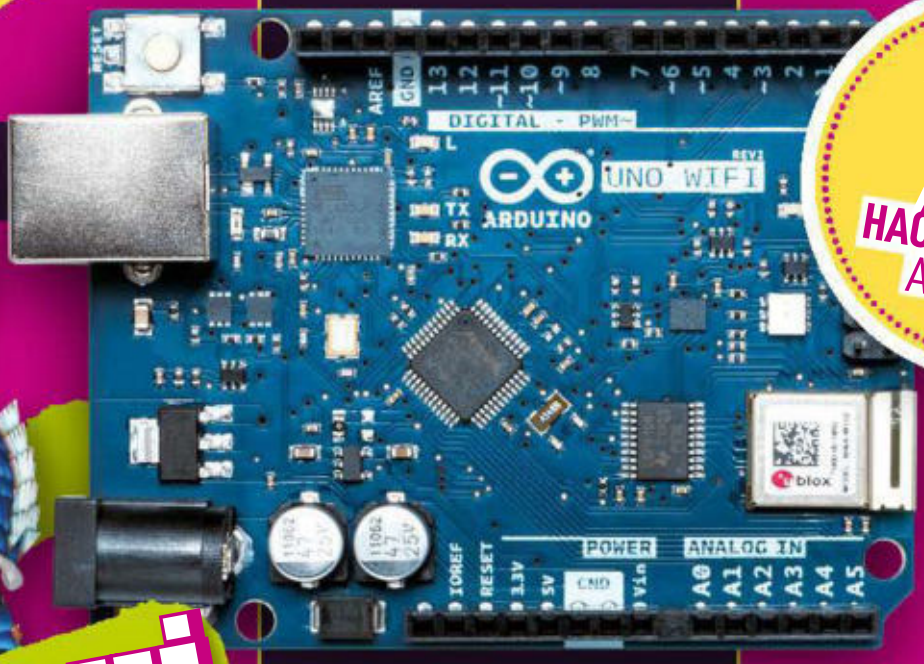
TECHNOLOGY IN YOUR HANDS

hsmag.CC | May 2020 | Issue #30

ARDUINO

WHERE CODING MEETS MAKING

DIY SMART SPEAKER



75 HACKS, MAKES, AND TIPS

LED ANIMATIONS



Kamui Cosplay

LOW-POWER RASPBERRY PI

May 2020
Issue #30 £6

9 772515 514006 30

3D PRINTING MAGNETS VACUUM FORMING MIRRORS

HELPING DRIVE YOUR INNOVATION



*A shipping charge of £12.00 will be billed on all orders of less than £93.00. A shipping charge of $18.00 USD will be billed on all orders of less than $50.00 USD. All orders are shipped via UPS, Federal Express, or DHL for delivery within 1-3 days (dependent on final destination). No handling fees. All prices are in British pound sterling or United States dollar. Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries. © 2020 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ECIA MEMBER



Welcome to HackSpace magazine

Like many people, I learned Arduino in a bit of a haphazard way. I played with examples, built projects, and learned as I went along. This was great fun and taught me a lot, but this style of learning is prone to leaving gaps in your knowledge. In this issue, we're highlighting some things that are really useful to know about Arduino and Arduino-compatible systems. Some you might know, some you might have forgotten and, hopefully, some you'll be learning for the first time.

This was great fun and taught me a lot, but this style of learning is prone to leaving **gaps in your knowledge**

The other thing we've been working on this month is helping produce protective equipment for health workers. I don't know what state the world will be in by the time you read this but, if it's still needed, and if you can help, take a look at page 74 to join in.

BEN EVERARD
Editor ben.everard@raspberrypi.org

PAGE 50
SUBSCRIBE TODAY



Got a comment, question, or thought about HackSpace magazine?

get in touch at **hsmag.cc/hello**

GET IN TOUCH

hackspace@raspberrypi.org
hackspacemag
hackspacemag

ONLINE

hsmag.cc

Available on the App Store

GET IT ON Google Play

EDITORIAL

Editor
Ben Everard
ben.everard@raspberrypi.org

Features Editor
Andrew Gregory
andrew.gregory@raspberrypi.org

Sub-Editors
David Higgs, Nicola King

DESIGN

Critical Media
criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Sam Ribbits, Harriet Knight, Ty Logan

Photography
Brian O'Halloran, Paul Carroll

CONTRIBUTORS

Lucy Rogers, Drew Fustini, Jo Hinchliffe, David Hunt, Mayank Sharma, Andrew Lewis, Rob Miles, Becky Stern, Gareth Branwyn, Les Pounder, Marc de Vinck

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.org

Advertising
Charlie Milligan
charlotte.milligan@raspberrypi.org

DISTRIBUTION

Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS

Unit 6, The Enterprise Centre,
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE

To subscribe
01293 312189
hsmag.cc/subscribe

Subscription queries
hackspace@subscriptionhelpline.co.uk

recycle

CC BY NC SA

This magazine is printed on paper sourced from sustainable forests. The printer operates an environmental management system which has been assessed as conforming to ISO 14001.

HackSpace magazine is published by Raspberry Pi (Trading) Ltd., Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISSN: 2515-5148.

Contents



124

06 SPARK

06 **Top Projects**
Creations made by humans, for humans

16 **Objet 3d'art**
A flawless time/shelving solution

20 **Meet the Maker:** Nick Hand
Keeping the flame of Gutenberg's craft alive

24 **Columns**
The make-do-and-mend mentality

26 **Letters**
Those brazing how-tos are starting to pay off

28 **Kickstarting**
CircuitBrains Deluxe: turn your project into a product

30 **Hackspace** DIYcave
Making in the Pacific Northwest

35 LENS

36 **Arduino**
The boards, the tools, the ABSOLUTE POWER!

52 **How I Made:** ISS tracker
Follow the International Space Station from home

58 **In the workshop:** Fixing Bluetooth
Resurrecting an ancient valve radio to stream Spotify

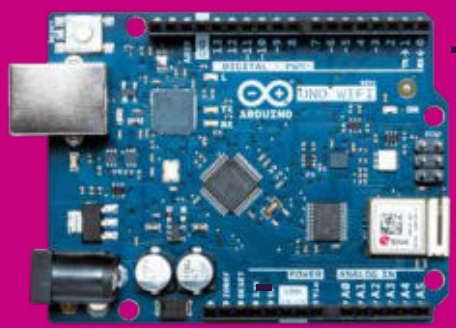
60 **Interview:** Svetlana Quindt
There are some serious skills in the world of cosplay

70 **Improviser's Toolbox:** Paper
Unique creations made of mashed-up wood-fibres

Cover Feature

ARDUINO

WHERE CODING MEETS HARDWARE



36

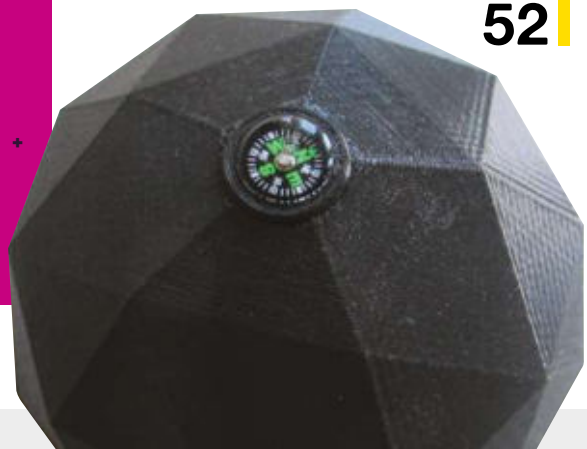
Tutorial

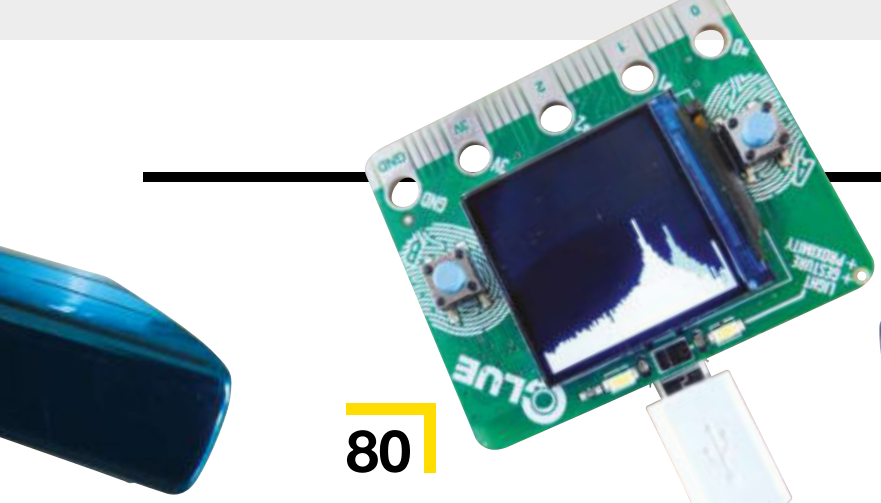
Fire in a jar



86 Gather around the warm glow of the LEDs for story time

52





80



118

Direct from Shenzhen

USB power meter



116 Test the power consumption of circuits for a bargain price



06



126

Interview

Svetlana Quindt



60 Electronics, programming, 3D printing… cosplay has it all

79 FORGE

80 **SoM** CircuitPython
Harness mathematical prowess with ulab

82 **SoM** micro:bit
Send data from one device to another with radio

86 **Tutorial** Fire in a jar
All the flickering ambience of real fire – with LEDs

90 **Tutorial** 3D-print wood
Get a more arboreal effect from your printer

94 **Tutorial** Magnets
Make life easier with these versatile attractors

98 **Tutorial** Low-power Raspberry Pi
Eke out battery life to keep sensors alive

102 **Tutorial** DIY vacuum former
Suck hot plastic into (almost) any shape

108 **Tutorial** Affirmation mirror
Start your day with positivity

112 **Tutorial** 3D printing supports
Experiment with options in Cura

115 FIELD TEST

116 **Direct from Shenzhen** USB power meter
Test current draw, battery strength, and more, cheaply

118 **Best of Breed**
The finest budget oscilloscopes on the market

124 **Can I Hack It?**
Explore the guts of a Kinect games controller

126 **Review** Adafruit CLUE
Many sensors on a tiny board

128 **Review** Kennedy workbench
Portable gripping power

Some of the tools and techniques shown in HackSpace Magazine are dangerous unless used with skill, experience and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. HackSpace Magazine is intended for an adult audience and some projects may be dangerous for children. Raspberry Pi (Trading) Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in HackSpace Magazine. Laws and regulations covering many of the topics in HackSpace Magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in HackSpace Magazine may go beyond. It is your responsibility to understand the manufacturer's limits.

IN-2 Binary Nixie clock

By **Marcin Saj**

hsmag.cc/k5xkt9

The old joke goes that there are 10 types of people: those who understand binary, and those who don't. Within that subset who understand binary, there's another, smaller fraction who can use it intuitively enough to be able to tell the time with it. It's this sliver of humanity for whom Marcin Saj has made this glorious binary Nixie clock.

The clock shows the time as binary numbers, showing the hour on the top row, the minutes in the middle, and the seconds on the bottom, which you can intuit if you spend any time watching it (or one of the videos that Marcin has put up here **hsmag.cc/juaJaq**).

Marcin sells these as add-ons for 5V/3.3V devices, which includes the Arduino Nano and Particle Photon. And as it's just a display (albeit a glorious one), you can also use it as a binary thermometer hygrometer with the right sensor.



Right
Marcin's clock uses a DS3232 real-time clock, just like the giant seven-segment clock on page 16



Speedometer clock

By Eamon Walsh

hsmag.cc/fkDDAV

This clock is a great example of how re-used components can be brought to life with a little bit of love. Eamon salvaged the speedometer and other dash instruments from a broken 1997 Volkswagen Polo (he says they came away easily, with just a cross-head and a Torx screwdriver – that's the star-shaped one). He made the steering wheel from two pieces of 18mm MDF glued together, and cut out the holes with a drill and a jigsaw.

There are actually two clock mechanisms here, both purchased online: the hours and minutes unit, housed in the speedometer, and the seconds hand, which is now in the rev counter.





Right
There's more detail on the build process at instagram.com/eamon_walsh_diy and facebook.com/eamonwalshdiy


mph
km/h
1/min x100
60
70
110

REGULAR

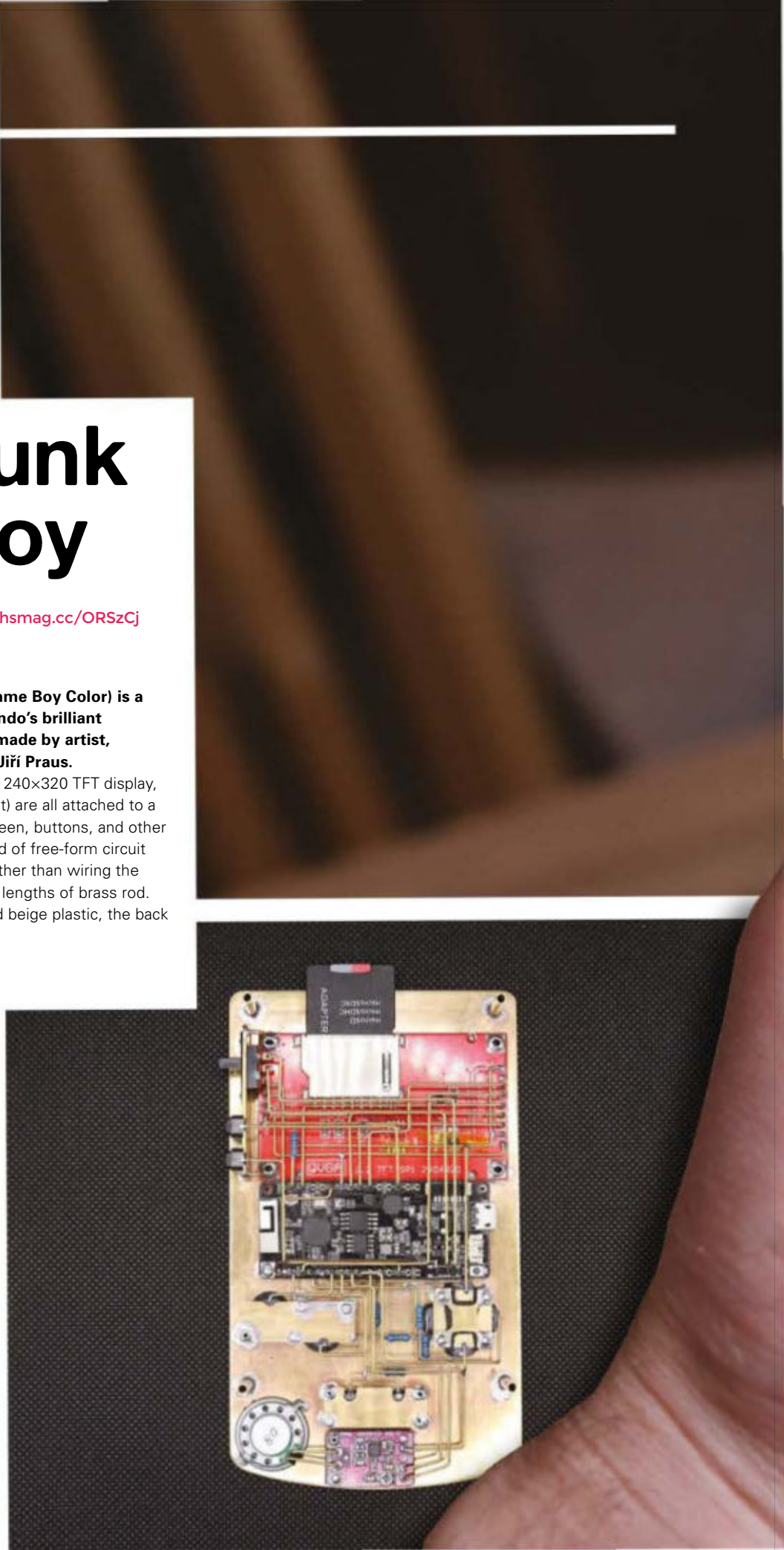
Steampunk Game Boy

By Jiří Praus

hsmag.cc/ORSzCj

This Game Boy (actually a Game Boy Color) is a re-implementation of Nintendo's brilliant handheld gaming console, made by artist, engineer, and programmer Jiří Praus.

The electronics (including a 240×320 TFT display, ESP32 board, and SD card slot) are all attached to a brass plate that Jiří milled to accept the screen, buttons, and other internals. Jiří's done a lot of work in the field of free-form circuit sculpture before, so it's no surprise that, rather than wiring the components up, he's connected them with lengths of brass rod. And rather than hide such neat work behind beige plastic, the back of the Game Boy is clear acrylic.



Right
As much as possible, this is handmade – even the 3D-printed buttons


THE LEGEND OF
ZELDA
AGES
PRESS START

Infinity table

By **Procupine**

hsmag.cc/JJP08b

A few mirror and lights can go a long way, and so it is with this infinity table made by Reddit user Procupine.

The frame of the table is made of oak; the LEDs are just three rows of WS281B LED strips, mirrored over and over into infinity and controlled by an Arduino Mega clone. We've seen simpler setups than this – most notably Angela Sheehan's infinity heels – but the advantage to incorporating an Arduino, and using RGB LEDs, is that Procupine can program animations, colour changes, or dimming/brightening effects into his living room furniture.



Right
There are not, in fact, infinite LEDs here: the effect is all light and mirrors



REGULAR



Right For a look at how makers are helping during the pandemic, turn to page 74



LED mask

By Chelsea Klukas

Lumencouture.com

Chelsea Klukas has been adding technology to fashion for over eight years, starting with one-off creations for fashion shows, and has recently moved into ready-to-wear retail with her Lumen Couture brand. Her latest creation is this mask, which was intended as a project to keep her busy over a weekend and has turned into a bit of delight that's been injected into a strange situation.

The mask incorporates a flexible LED panel – controlled via Bluetooth – which displays text, images, and (our favourite) an audio equaliser graphic.

There's a pattern for the mask at **wearabletutorials.com**, where you can also buy the flexible LED panel. The electronics components can be removed completely for cleaning, and swapped with a regular air filter to be dual-purpose as a practical safety mask.





REGULAR

Objet 3d'art

3D-printed artwork to bring more beauty into your life



This project, by Lewis Aburrow, caught our eye because it's not just a 3D-printed version of something else; it just wouldn't be as good if it were made any other way. Apart from the electronics and the back-board (Lewis's design calls for a wooden board at least 112 cm x 39 cm), everything about this is 3D-printed.

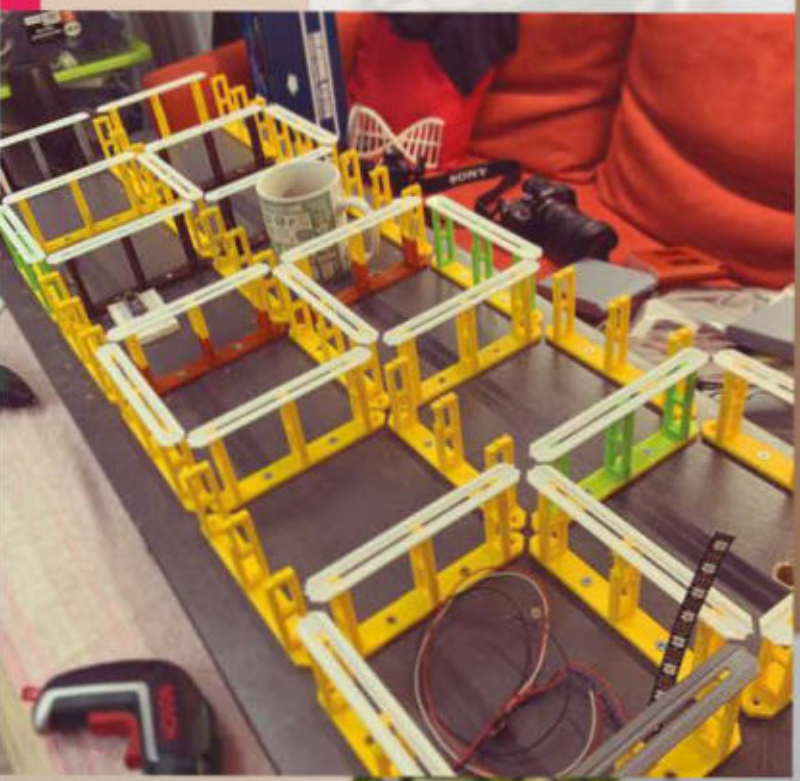
It's a seven-segment clock that's also a shelf unit, with in-built LEDs to display your precious things. It even uses a photosensitive resistor module to dim the lights of the clock when the room goes dark. The maker stalwart, NeoPixels, give RBG programmability to the light effects, so you can control them any way you wish – the whole thing is controlled by an Arduino Nano and a DS3231 real-time clock. →



Above
This clock uses an opulent 5m of NeoPixels

The core supports – providing the internal framework and support for the LEDs – are printed at a layer height of 0.2mm, and as they're hidden from viewing the final build, there's a good way to use up any bits and bobs of filament that you may have lying around. To ensure you attach these to the back-board with consistent spacing, Lewis has released a file for CNCing the pilot holes for the wood-screws, or if you don't have access to a CNC machine, he's also made a 3D-printed jig to guide your drill.

The first couple of layers of the sleeve are printed in white; Lewis then switched to a wood-infused PLA (for more on the ins and outs of using wood-infused 3D prints, turn to page 90). The benefits of using two types of filament for this part are that the white layer provides diffusion for the LED strips in the seven-segment displays, while the wood layers prevent light leakage and give a more natural finish. If you've got 18 minutes to watch an expert talking with a measured blend of enthusiasm and authority, he's done a build video here: **hsmag.cc/6cLc37.**



Below
These internals are hidden by woodfill



REGULAR

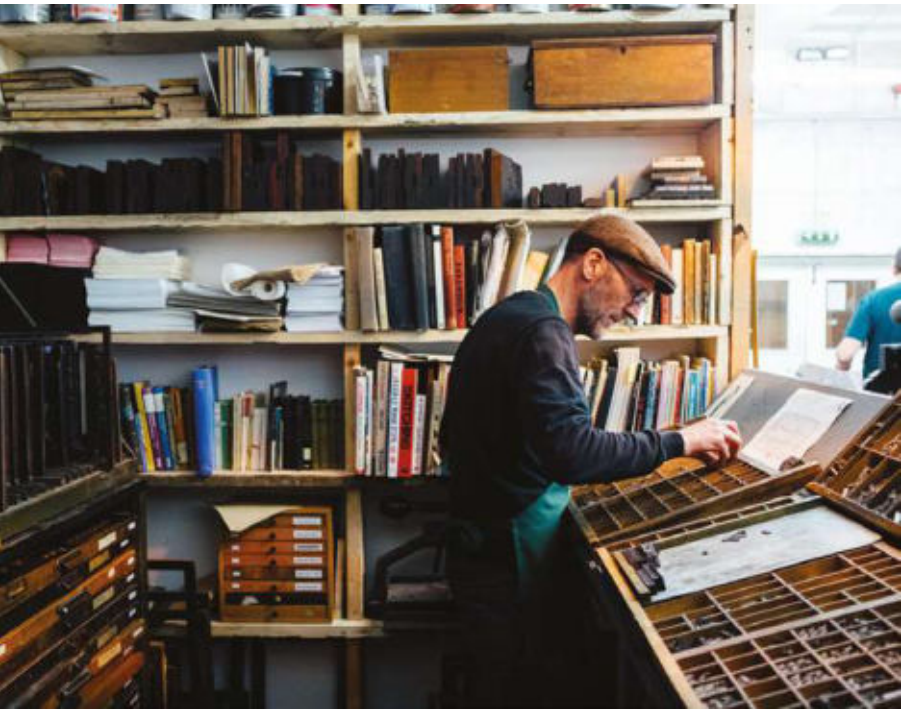
Meet The Maker: Nick Hand

Keeping the craft of metal and ink alive



Image Jonathan Cherry

We're used to getting information in electronic form from seven-segment displays, OLEDs, VFDs, Nixie tubes, even single LEDs blinking on and off. But before all of these technologies, humans sent messages to one another by putting ink on paper using bits of metal – and if you're reading this in print, you'll know that we still do.



Below Nick at work in the Department of Small Works, Bristol

The craft of letterpress is still alive and going strong. It's no longer the primary way of transmitting the written word, but it's out there, and it's still extremely lovely. One of the makers keeping the craft alive is Nick Hand, who makes beautiful things at The Department of Small Works in Bristol, England. We spoke to him from his studio in the old part of the city, and could almost smell the ink down the telephone line.

"I trained as a photographer in the 1970s and, at that time, there were no computers in graphic design or typography. I worked in a room with letterpress machines, with wood type and a couple of technicians. It was very unionised, so you had to have a technician for print and a technician for typesetting – they were seen as two different trades then.

"I had a career as a graphic designer and, about halfway through that career, Apple brought in desktop machines. About ten or twelve years ago, I got a bit bored with sitting in front of a screen, so I started going on bicycle rides. Ten years ago, I rode around the coast of Britain, and I met about 130 craftspeople along the way. I interviewed them and recorded them talking about their craft, and edited the interviews into these little films in a tent at night, so I could get them done the same day.

"I did the same around the coast of Ireland, which was another 1800 miles, so it was about 6500 miles in total. That experience opened my eyes to the importance of craftsmen and women, the skills that they have, and the legacy of craft. It also made me fall in love with cycling. I was a bit scared at first that if I went off cycling for three or four months, I'd lose focus – but if I had a task, which was recording these people, it became a job that I could focus on.



"I carried on recording craftspeople. And at some point, I remembered being back in that room with the letterpress machines and thought I'd find out what was happening in Bristol. I found some type and machines in an industrial museum called M Shed. The curator there told me he'd much rather it were being used in a way that's visible to the public, rather than tucked away in a store-room. So he let me have some stuff on a long-term loan, which we still have seven or eight years later.

"Then we found a space that was part of a co-op in the middle of Bristol, which means that the rent is quite affordable for a decent-sized space, and we've gradually built on that. We've collected more equipment and more type, a little library of books, and we've become quite a nice resource for letterpress in Bristol. We have workshops that people come to, evening classes, and open studios where people can come in and print their own projects. We also print things ourselves and sell them: posters, playing cards… it's like a playground of creativity.



Above
Nick runs classes in his studio for lovers of ink and metal

THE LETTERPRESS PROCESS

"First of all, you choose what fonts you want to use. It's like choosing fonts on a computer, but they're physical fonts, so you have big cabinets full of lead type and wood type, and you decide which you want to use for your project. The fonts are laid out in a functional way a bit like a qwerty keyboard. You collect the type in something called a composing stick. →

REGULAR



Above
Maybe we're biased, but there's something beautiful about quality print work

"Once you've got your letters together, you lock it in place and combine it with some images perhaps, which would be cut into lino or wood, or etched into metal. We use small bits of metal to space out the characters – which is where you get the term leading in typography – and also bigger bits of wood or metal, called furniture, to space out the elements.

"You lock all that together in what we call the chase, which is the metal rectangle that fits on to the press. Once it's locked together, you call it a forme (with an 'e' on the end). You ink that forme up, then pull a proof, which is one sheet that lets you check for errors.

"When you set the type, it's always upside down and back to front, so you can imagine it's very easy to make mistakes. Once you've corrected it, you're ready to go: choose the paper, the ink, the colours, and then you print. Generally, you print on a larger sheet of paper and then trim it afterwards. Then you'd bind it, either mechanically or by hand.

> When we do workshops, the place is always very quiet **– you get lost in the process**

"Typesetting is very therapeutic. When we do workshops, the place is always very quiet – you get lost in the process. There's a legacy to type. You can feel the marks on it, you know it's been used many times before. You're aware that you're handling something old and important.

"I was speaking at an event a while back, listening to someone talk about MakerBot, a 3D printer that you started by printing itself. It was a few years ago now, but when he was speaking I remember thinking that he's crafted that product; he'd gone through all these iterations. After it was my turn to talk, someone in the audience said to me, 'It's so nice to see proper craft, rather than this modern 3D printing stuff.' I didn't agree. I think they're both crafts; they're just different crafts.

"A lot of people are bringing more electronics to letterpress as well, which I find fascinating. The type I use is either made from molten lead or carved out of wooden blocks. People have been using laser cutting and 3D printing in flexible filament to make their own custom letter sets. You have to be open to mixing technologies all the time.



THE GERMAN CONNECTION

"The oldest machine we have is an 1832 Albion press. If you saw the film *Peterloo*, there's a similar press used in that. It's a similar-looking press to the one that Gutenberg would have used in 1450, except that his was wooden. We call it a vertical platen press. Our one came from about two streets away, so it's travelled a few hundred yards in 190 years. It's very beautiful, and it works as well as the day it was made. The Albion presses have a great Victorian aesthetic; they have lion's feet and a crown. There's an American version called the Columbian press, which has an eagle on top. It feels like they were competing for the best symbols.

"Letterpress has become popular again as a craft. There is one company that has started casting the parts for an Albion press again, so you can buy one new; we know that because the one we have has had bits missing, and we were able to buy parts from this company. They're made of cast iron, so I guess they make moulds from an old press and cast them from that.

"I had a friend here who made bicycles. It became a bit of a challenge to see whether he could make a bicycle to put a printing press on. He made a lovely bike to put an Adana 8×5 printing press on the back. While he was making it, we decided to go on a journey. We thought the most appropriate journey would be to go from here to Mainz in southern Germany, which is where Gutenberg invented printing with movable type. The Chinese were printing a couple of hundred years before, but he made the big breakthrough of melting lead into a little matrix and making multiple letters, and locking them together to be able to print multiple configurations of letters.

"It was a massive invention for changing the world forever. I just wanted to do something to honour the invention".



Below Heidelberg presses are still in use to this day as a conveient way of punching patterns in paper

Make-do-and-mend

Keeping things running



Lucy Rogers

@DrLucyRogers

Lucy is a maker, an engineer, and a problem-solver. She is adept at bringing ideas to life, and is one of the cheerleaders for the maker industry.

As makers, we often use the 'make-do-and-mend' mentality. If something is broken, many of us will try to fix it. If that doesn't work, we may try to adapt it. If that doesn't work, then we will probably cannibalize the thing for parts for future projects.

I have many boxes of 'parts' just waiting for the right problem, which has often been very useful when I have left a make to the last minute and don't have the time to order in new parts.

I often use 'workarounds' too. I don't have the exact thing I want to do this, but if I use this, this, and that, it will work. Which is why some of my projects look a little like a Heath Robinson contraption.

Maintenance is something that is rarely the first choice for me. I prefer the buzz of fixing something that isn't working, rather than maintaining something that is still OK for now. But the old saying 'a stitch in time, saves nine!' doesn't get any less true with age. Having become more aware recently that not everything is available on 'next-day delivery' – and having to be more conscious of where my cash goes – the make-do-and-mend mantra also makes economical and environmental sense. And so, I am paying more attention to the things that need preventative maintenance – before they get to the 'not working' stage.

For example, that slight drip on the showerhead? I've fixed it with Sugru. I've oiled my bike chain for the first time in years, and I've sewn up that loose thread on my jacket. I've been surprised at the delayed satisfaction I got from these projects. I hadn't really realised that every time I saw that thread and drip, I thought 'must fix that'. And my bike rides are easier now! I may not get the immediate buzz, but the cumulative effect on my mental health over time more than makes up for it.

> **I am reducing the number of things that I buy – which is hard, because 'ooh look, shiny!'**

I am also following the 'reduce, reuse, and recycle' mindset. I am reducing the number of things that I buy – which is hard, because 'ooh look, shiny!' I am reusing the things that can be reused – I have a towel that was used by me, then relegated to be a dog towel, and now it's a rag in the workshop. I do love rag rugs too, so may start making one of those soon from other bits of rag. And I recycle whatever I can.

What projects are you doing that make-do-and-mend?

PCB love - making circuit boards

Turn your temporary projects into permanent fixtures



Drew Fustini

@pdp7

Drew Fustini is a hardware designer and embedded Linux developer. He serves on the board of directors for the Open Source Hardware Association and the BeagleBoard.org Foundation, and is an ambassador for the RISC-V Foundation. Drew designs circuit boards for OSH Park, a PCB manufacturing service, and maintains the Adafruit BeagleBone Python library.

You made your first LED blink, you learned how to use a breadboard, and you know which end of the soldering iron to hold. So what's next? High up on the list of essential electronics skills is learning how to design and make your own printed circuit board (PCB). My first PCB was a POV (persistence of vision) display, a modified version of an Adafruit open hardware design. I still remember how exciting it felt to open up my mail and solder a board I designed myself!

A PCB is a board with traces (lines) that connect different pads or holes for electronic components to each other, or to connectors that allow us to hook up power, microcontrollers, or other parts. We use solder to connect the pads or holes on the bare board to the component that the board was designed for. Once the components are in place, the traces let us push power or data around our boards.

PCBs themselves are like a sandwich made up of layers of different materials. The main 'body' of a PCB is made of some kind of substrate; typically a fibreglass called FR4. On top of that is a thin layer of copper-foil that lays out the traces and pads, then a layer of coloured solder mask which prevents solder from sticking where it shouldn't. This solder mask is often green, but you can get all sorts of colours, such as my favourite – purple. The copper that is not covered by solder mask then gets protected by a surface finish such as HASL (solder) or ENIG (gold plating). Finally, there is the silkscreen layer where text and symbols are printed.

Like any good sandwich, you can keep adding in more layers of copper, from two layers up to tens of layers. It's also worth knowing that rigid fibreglass isn't your only option: you can choose a material called Kapon (polyimide film) to make flexible PCBs. You can even make your PCB sandwich using both fibreglass and Kapon to make a board that is flexible in some places and rigid in others. One super-cool option, that both OSH Park and Evil Mad Scientist Laboratories have been using recently, is black fibreglass paired with a clear solder mask, so all your copper traces are visible.

> **One super-cool option is black fibreglass paired with a clear solder mask, so all your copper traces are visible**

In next month's column, I'll be talking about the awesome ways the hardware hacker community has been using PCB design to make cool electronics projects and beautiful art, as well as sharing some of the tools they're using to make their boards. In the meantime, I recommend this excellent video that visualises the composition of a PCB: **hsmag.cc/SI2m31** (*How do PCBs Work?* by Branch Education).

Letters

ATTENTION ALL MAKERS!

If you have something you'd like to get off your chest (or even throw a word of praise in our direction) let us know at **hsmag.cc/hello**

PHONES

Justine Haupt's rotary phone… it's exactly what I want. No bells or whistles, no screen that gets greasy/breaks/glows fierce blue light when I just want to know what time it is in the morning. Genius. I hope she makes a million.

James
Columbus, Ohio

Ben says: I thought so too, but then we discovered the calming effect of playing *Baby Shark* to an angry toddler. I'd struggle to give mine up, but if I ever did, I hope it would be with something like Justine's incredible creation.





PPE

With all that's going on, I've been impressed by all the 3D printer owners coming together to make face shields for healthcare workers. This is the perfect test-case for 3D printing – rapid prototyping, throw out version 1, make a start on version 2, and get on with it. Now the makers have done the hard work, we just need a company with spare injection moulding capacity to step up and speed up the manufacturing process.

Paul
Leeds

Ben says: The mass mobilisation of makers to 3D-print protective equipment has been wonderful to see. There are more projects going on than we can give credit to, but each one is making a positive difference.

PRAISE!

Whatever Dr Andrew Lewis wants to write, please just let him write it. I followed his advice from issue 23, and brazed together a chimney for a meat smoker out of tin cans and, now the weather's picking up, I'm getting my money's worth out of it. It looks a bit *Beverly Hillbillies*, but I think the homemade aesthetic just adds to the flavour.

Dan Armstrong
Reading

Ben says: Dr Lewis is indeed on a rich vein of form at the moment. And, staying on a culinary theme, he's turned some roasting tins into a DIY vacuum former this issue.





SHEDS

I was sceptical when I read the first bit of the interview with the chap from Men's Sheds (Charlie Bethel, in issue 29). Safe spaces? Aren't they for students who can't handle the real world?

When I got a bit further, I started to get the point. You need somewhere you can switch off and be yourself. It's not about cutting yourself off from the real world; it's about giving yourself permission to focus on one thing at a time.

Tom McGee
York

Ben says: We were impressed by the ethos of Men's Sheds too – the woodworking drew us in, but we know so many people who switch off at the end of the day by bashing bits of wood together, or coming up with a design in Tinkercad, or swapping bits in and out of a circuit. More power to them.

CROWDFUNDING NOW

CircuitBrains Deluxe

Turn your project into a product

From $29 | **crowdsupply.com** | **Delivery:** July 2020

So, you've created a great device using an off-the-shelf microcontroller and want to build it into a circuit board, either for yourself or a product. What do you do? You could design the PCB from scratch, manufacture it, and solder it all up, but this is quite an involved process, both in terms of design and manufacturing (as many modern microcontrollers have teeny-tiny pins that are a challenge to solder). It can be done, but if you want to spend your time on your product, rather than your PCB, it may not be the best use of your time.

> "It's ready to help you turn your **CircuitPython project into a product**"

The solution to this is to use a module that wraps everything you need into a PCB that can itself be soldered straight onto your larger PCB. This only works if the module has no components on the bottom, and pads for soldering. Fortunately, the CircuitBrains Deluxe has just this. It's also got a SAMD51 microcontroller, 8MB of flash, and comes preloaded with CircuitPython, so it's ready to help you turn your CircuitPython project into a product.

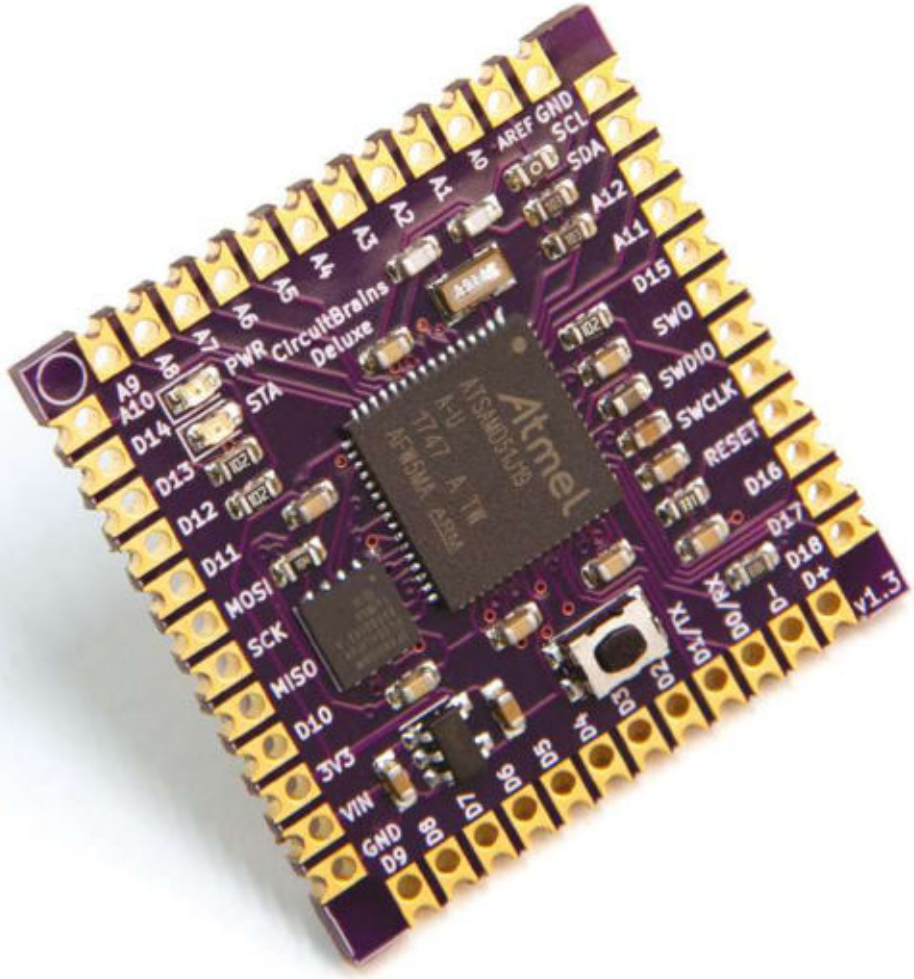
There's even on-board power circuitry, so you can supply it with 5V (such as via a USB connection), and it will handle the rest, just as you would expect a development board to.

There are Eagle and KiCad libraries, so if you're using one of these PCB design tools, you can just bring the CircuitBrains Deluxe in, and it will create a part with the correct pins and footprint on the PCB.

Modules like the CircuitBrains aren't for everyone. If you're doing large production runs, or are used to working with the chips needed, they may not be cost-effective. However, they massively lower the barriers of entry to creating custom boards for your project. If you're coding with CircuitPython, and are looking to build your project into a board, then this is something you should take a good look at.

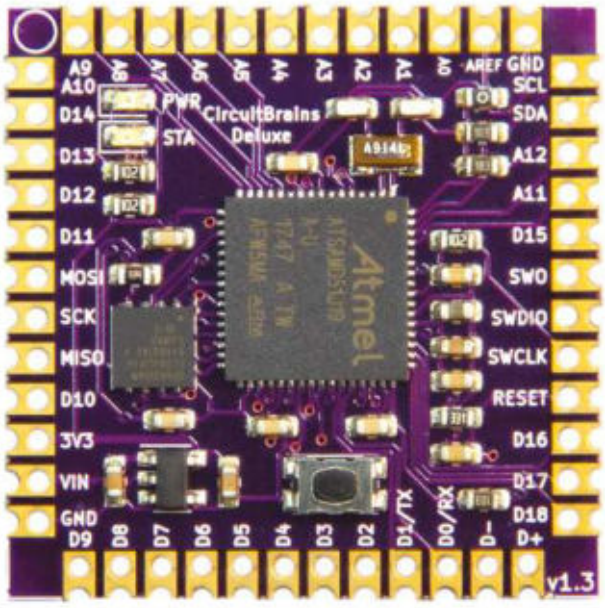
BUYER BEWARE

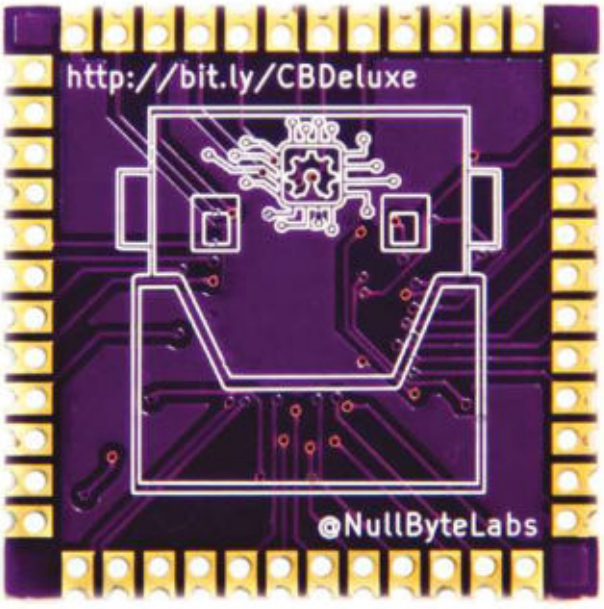
When backing a crowdfunding campaign, you are not purchasing a finished product, but supporting a project working on something new. There is a very real chance that the product will never ship and you'll lose your money. It's a great way to support projects you like and get some cheap hardware in the process, but if you use it purely as a chance to snag cheap stuff, you may find that you get burned.



Above
The SAMD51 is a powerful processor

Below
There are no components on the bottom, to let it sit flat on a PCB





Space of the month: DIYcave

By **Annie Willis**

DIYcave

diycave.com

DIYcave

@DIYcave

DIYcave

Aaron Leis, Dave Danek, and Tim Willis shared a common dream. They wanted to create a makerspace where people could share tools, equipment, ideas, and expertise. When they met in 2013 at the Bend Fall Festival, they knew they were meant to bring this idea to life.

DIYcave opened in an old mill building in central Bend, Oregon in spring 2014. Here, staff teach community members skills that they may not otherwise have a chance to learn, and help others hone forgotten skills. Everyone from school-aged children to adults can take advantage of an array of classes and creative opportunities for self-expression or entrepreneurship at DIYcave.

DIYcave is a makerspace that bridges tech to handmade, and prototype to fine art. We focus on providing as many classes as possible to teach

> "DIYcave functions like a health club. Instead of exercise equipment, **we have tools and expertise**"

those in our community who may be unfamiliar with the 'Do It Yourself' movement. Our fantastic teachers have helped many new makers feel that they have the skills and confidence to bring their dreams to life. While many makerspaces focus mainly on computing and technology, DIYcave also makes room for the industrial arts, such as wood crafts, metal arts and tooling, large sculpture art, and welding, as well as automotive repair, and large project builds such as tiny homes and van conversions. The balance of all these disciplines, →





Above
Keeping old items functioning is a crucial part of many makerspaces









as well as tech and prototyping, makes DIYcave unique among all makerspaces.

DIYcave functions like a health club. Instead of exercise equipment, we have tools and expertise. Membership allows access to the tools and workshop space to work on a maker's dream projects. We provide the equipment and space; members supply the materials and hardware needed for their projects. And we have experts in many fields available to advise members on whatever project they have in mind, from a friendship bracelet to a tiny home!

For those who just want to dabble, but not commit to a membership, classes are available to all. No membership is needed. Community members can learn to weld with a partner during Date Night Welding, craft jewellery, or try their hands at building a pallet wood shelf. Our teachers are happy to hold special classes for a group that would like to learn together. For the more serious makers, we offer industrial/art space for rent, and that is next door in the DIYcave's Annex, which is also a creative, co-working environment.

DIYcave offers a unique opportunity for those who want to volunteer. Our shop is run almost entirely by volunteer shop stewards, who receive access to the shop and classes in their free time to tackle their own creative pursuits.

"Every day is a new creative horizon. There is always a little more to learn, a little more to try. "
– Aaron Leis, Founder. □











CONTACT US

We'd love you to get in touch to showcase your makerspace and the things you're making. Drop us a line on Twitter **@HackSpaceMag,** or email us at **hackspace@raspberrypi.org** with an outline of what makes your hackspace special, and we'll take it from there.

Above
Igniting a burning passion for making

Left
Forging new friendships next to the, err, forge



Build a Makerspace for Young People

Join our **free online training course on makerspace design** to get expert advice for setting up a makerspace in your school or community.

Sign up today: rpf.io/makerspace

Raspberry Pi

The Raspberry Pi Foundation. UK registered charity 1129409

#RPiLearn

HackSpace
TECHNOLOGY IN YOUR HANDS

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

PG 52



HOW I MADE: ISS TRACKER

Follow the International Space Station around the skies with homebrew engineering

PG 58

IN THE WORKSHOP

Upgrading the innards of a vintage valve radio-cum-Bluetooth speaker

PG 60

INTERVIEW: SVETLANA QUINDT



Be amazed by the incredible multi-skilled talent that goes into building cosplay



PG 36

ARDUINO

WHERE CODING MEETS HARDWARE

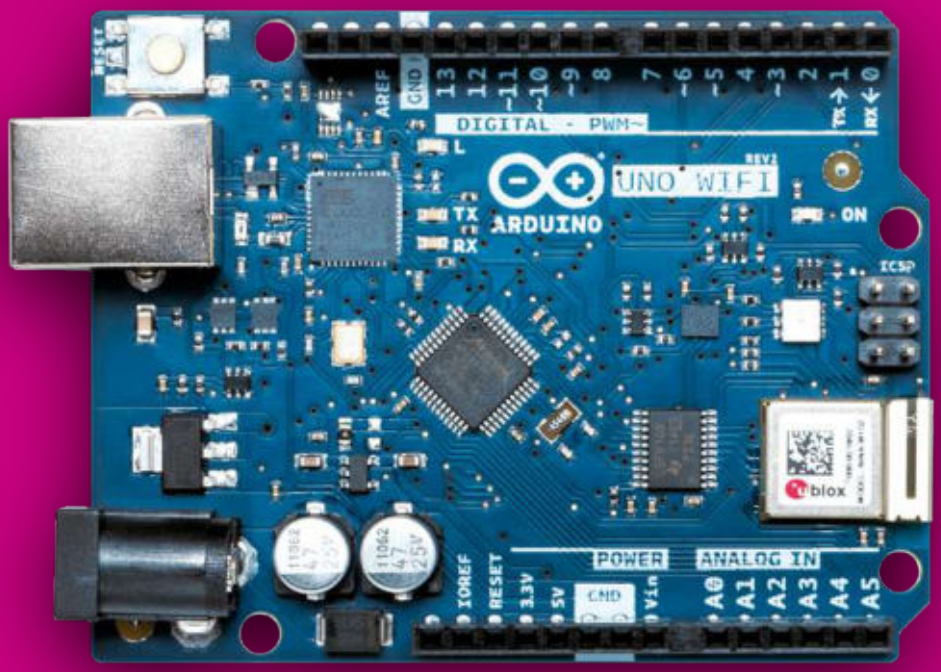
PG 70

IMPROVISER'S TOOLBOX: PAPER

Make unique things with thin sheets of mashed-up wood-fibre

ARDUINO

WHERE CODING MEETS HARDWARE



We're living in a golden age of microcontrollers. There's a wealth of control boards available to suit almost any budget and requirement set, and a huge range of hardware that can be easily hooked up to them. Thanks to the Arduino ecosystem, we also have a framework for linking all this together.

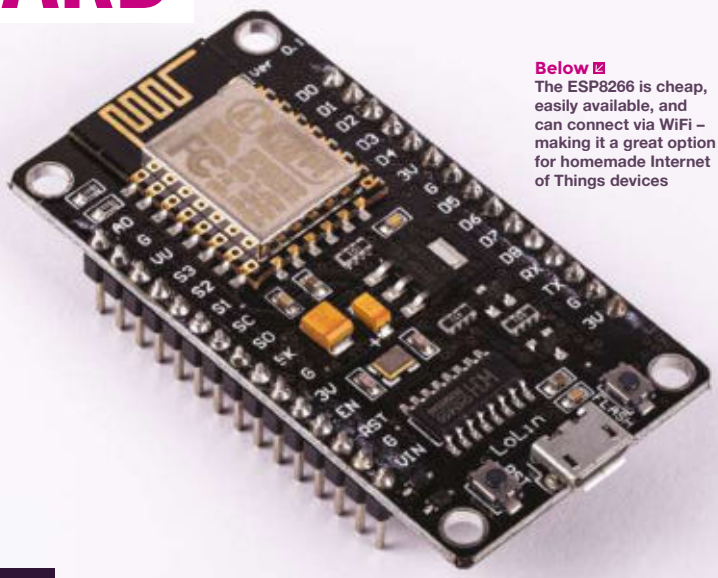
The downside of this embarrassment of riches is that it can sometimes be hard to find your way through all the possible boards, hardware, and libraries that exist. Which controller should you use? What libraries can make my programming easier? How should I power my project? Just finding the right place to start answering these questions can be a daunting task, but we're here to help.

Let's start with tip number 1: If it works, your project is good. It can be easy to compare yourselves to others who have different skill sets, time and material budgets, and requirements for their projects. It can be really inspiring to see great projects that other people have created, but remember that it's not a competition to create the fastest, smartest, and most beautiful project. Your project might have a bodge or two and be held together by glue (spoiler alert: most projects are), and that's great. It's yours, and it's perfect in its own way. Each project, big or small, is part of your evolution as a maker: part of your journey. What's important is that your maker journey is going in the direction you want it to.

How do you take your maker journey in the direction you want to go? Well, we've got another 74 tips to help you discover that. Let's find out! →

15 WAYS TO PICK THE PERFECT ARDUINO BOARD

There is a huge range of **Arduino-compatible boards available.** Some from huge global manufacturers, some from small, local firms, and even some that you have to build yourself. Which should you pick for your next project? It's impossible for us to rate every one individually, but here are our top tips for deciding which is most appropriate for you.



Below
The ESP8266 is cheap, easily available, and can connect via WiFi – making it a great option for homemade Internet of Things devices

5 FORM FACTORS

The shape of the board affects what hardware expansions you can use, what enclosures you can fit your project in, and how many features the manufacturers can pack on. Here are a few of the most popular form factors for Arduino-compatible microcontrollers.

1 **Uno** This is the classic shape that many people associate with Arduino. It's quite big by modern standards, but that chunkiness can make it easy to handle for beginners.

2 **Mega** This is an expanded version of the Uno to give lots of space for input and output pins. If you need lots of control, this could well be the option for you.

3 **Feather** A smallish footprint that now has a wide variety of add-on boards (known as Wings) available.

4 **ESP** There are lots of similar, but not quite the same, form factors used by ESP boards, so make sure you're getting the right one for your project.

5 **Large Pads** There are a few variations on this form factor, but they tend to be round and have large holes for connections. Originally popularised for wearable projects, these have become more common because they're easy to connect with crocodile clips.

6 ARDUINO-COMPATIBLE MICROCONTROLLER FAMILIES

The chip at the heart of any microcontroller board will define much about how it's used, and what it's useful for. Here are six of the most popular microcontrollers (and families of similar microcontrollers) for Arduino-compatible boards.

1 **ATmega** There are a few microcontrollers based on the AVR instruction set found on classic Arduino boards, such as the Uno and Mega. They're well tested and have served a generation of makers well. However, they aren't as fast or feature-packed as some other options.

2 **SAMD21** This series of chips are faster than the ATmegas, while still being small and reasonably priced – it's a great choice for general-purpose projects that don't have many specific requirements. An ARM Cortex-M0 processor core sits at the heart of the SAMD21 microcontrollers.

3 **SAMD51** Bigger and faster than SAMD21 and comes with more I/O options, the SAMD51 is based on the ARM Cortex-M4 and can be run at a high clock-speed (sometimes up to 200MHz). It's a great choice if you need processing power or want to control lots of hardware.

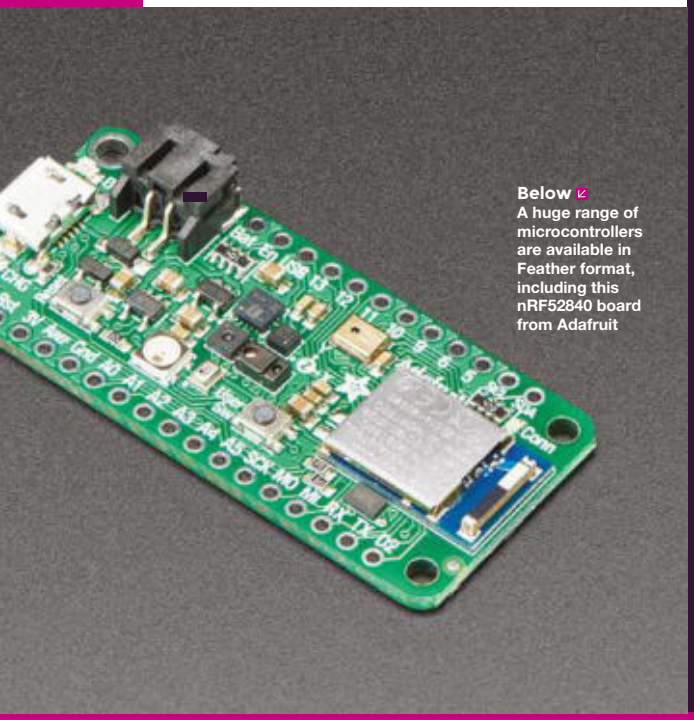
4 **nRF52840** This ARM Cortex-M4 microcontroller series is a bit less powerful than the SAMD51 and doesn't have as many I/O options, but it comes with integrated Bluetooth for short-range wireless communication.

5 **ESP8266** This revolutionised the microcontroller world when it came out, as it includes on-board WiFi and can cost only a couple of pounds.

6 **ESP32** The grown-up version of the ESP8266 represents a significant upgrade, particularly with regards to encrypted communication and processing power, yet it is still a very affordable option. →



Above Official Arduino boards originally used ATmega chips, but some now – such as this MKR WiFi 1010 – use SAMD21 microcontrollers



Below A huge range of microcontrollers are available in Feather format, including this nRF52840 board from Adafruit

FOUR TYPES OF MEMORY

Arduino-compatible boards will have several types of memory in varying amounts. Understanding what these different types of memory are will help you understand whether a board is suitable for your project.

1 **Flash** This is where the program is stored. When you compile your sketch, it will tell you how much flash space is needed to store it.

2 **RAM** This is used for variables your program uses. If you stream data from the internet, or control a lot of addressable LEDs, you'll need a comparatively large amount of RAM.

3 **EEPROM** This is usually a small amount of memory that you can write to, and is saved between restarts. It's useful for holding things like the identifier of the board if you have lots of boards running the same code.

4 **SPI flash** This is external memory that may be included as an extra chip on the microcontroller board. It can be used to store things like images or sounds (or, in the case of CircuitPython, code).

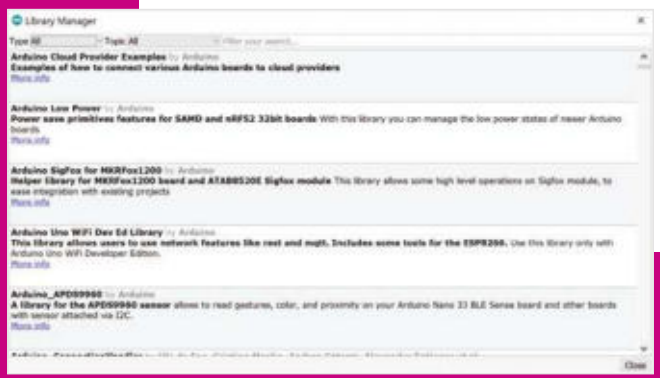
TOP 10 LIBRARIES IN ARDUINO

How do you pick the top libraries in Arduino when there are so many to choose from? We've taken the community route, and here are the ten with the most stars on GitHub (see **arduinolibraries.info** for other top libraries). They really show off the range of different things that people are using Arduino boards for.

Not all of these will work on all boards, so depending on your hardware, some will be more useful than others.

1 ArduinoJson JavaScript Object Notation is a way of structuring data in text files to make it easy for computers and people to read. It's hierarchical, with objects being held in curly braces, and properties within them being defined in the format 'name:value'. While this was originally used for JavaScript, it's language-agnostic. Many web-based data sources offer JSON formats.

2 WiFiManager Managing WiFi networks is a problem on embedded devices where you don't have a keyboard and screen to enter login details. This library works on the ESP8266 and will connect to any WiFi network it knows about. If it doesn't find one, it'll create its own WiFi network that you can connect to and, on there, there's a webpage that you can enter details of which network you'd like it to connect to.



Right The Arduino library manager makes it easy to download and install new libraries on your machine

3 FastLED A library of efficient routines for handling large LED displays. See the LED page (42) for more details.

4 IRremote Infrared controls might not be as fancy as WiFi controls or Bluetooth, but there's a whole lot of devices out there which already have an IR port. This library lets you put an infrared LED on your Arduino and send messages to a huge range of devices. You can use this library to turn off TVs, control your stereo, and much more.

5 Blynk Design a mobile phone app for your project (free for non-commercial use). See the IoT section (page 44) for more information.

6 PubSubClient This library connects to an MQTT broker (see IoT section) to send and receive messages.

7 PJON The Padded Jittering Operative Network is designed to work between devices without the need for any form of master server to control communications. It can work over different transports (such as Ethernet, serial, or LoRa), and works on a wide variety of hardware.

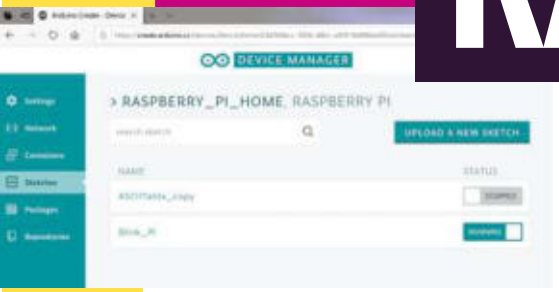
8 Adafruit NeoPixel Make your projects shine with RGB LEDs (see the LED page for more details).

9 WASM3 Web Assembly (WASM) is a subset of JavaScript that's designed to run far more efficiently than the full version. What makes it particularly useful is that compilers for many languages can output WASM, so it's become a popular byte-code language. This library allows you to run a WASM interpreter on an Arduino, which means that you can run semi-compiled code from these languages on embedded microcontrollers.

10 MFRC522 Near Field Communication (NFC) is sometimes known as contactless (which is a little confusing because it almost always uses a contact or 'tap' to get it close enough for the data transfer). In essence, it's a very short-range radio protocol, designed to work with one of the devices not having power, so the communications signal itself can power one device. MFRC522 is a popular series of NFC devices, though others are available. They work well for identity tags (think key card access).

ARDUINO ON RASPBERRY PI

RUN SKETCHES ON YOUR LINUX DEVICE



Below Run multiple Arduino sketches on your Raspberry Pi from the Arduino Create Cloud

It's not very well known, but you can use the online Arduino Create Cloud to program and control your Raspberry Pi (or other Linux-based devices). To get started, you need a Raspberry Pi connected to the internet and an account at **create.arduino.cc**. Log in to the Arduino cloud, go to Device Manager, scroll down to My Linux Devices, and click on Add New Device.

You'll need either to have a screen, mouse, and keyboard plugged into your Raspberry Pi, or another computer connected to the same network as Raspberry Pi. The latter is easier and a more likely setup for a home network, so we'll look at that.

You'll need to know the IP address of your Raspberry Pi. IP addresses are numbers assigned to every computer on the internet so that bits of data end up in the right place. They can get a little confusing if you're not used to how networks work. A computer connected via a home network to the internet will have two IP addresses – one internal to the local network, and one external to the internet. We need the one internal to the local network, and the easiest place to find this out is on your router. There's guidance on how to do this at **hsmag.cc/gMtt44**.

If this fails, you'll need to plug a screen and keyboard into your Raspberry Pi – details of how to find the IP address from the terminal are also given in the previous link. Once you've got the IP address, you can enter it in the webpage, and then click through to set up your Raspberry Pi. You'll end up at the device page, which lists a range of things about the device, such as the available memory and disk space, and you can control software running on your computer in three ways:

Docker containers These are ways of packaging software so that it should work on a wide range of different devices. It's a complex subject that we won't get into here, but details can be found at **docker.com**.

Arduino sketches These are exactly the same sketches that you create to run software on Arduino boards. In the Sketches tab, you can click on Upload a New Sketch, and you'll be taken to the Web Editor. This is very similar to the Arduino IDE and lets you create sketches that you can then upload automatically to your Raspberry Pi. You can run multiple sketches at once, and on the main Sketches tab, you can select which ones you want to run at any time.

Package manager Here you can install software, just as you would with the APT package manager on your Raspberry Pi. →

LIGHT UP THE WORLD

7 WAYS TO GET THE MOST OUT OF LEDS

1 **Dimming** Most Arduino-compatible boards can't vary the amount of current they output to an LED, so the brightness is fixed by which resistor you use. However, you can turn the pin on and off very fast, and to the human eye, that looks like making it dimmer. Pulse-width modulation (PWM) is a hardware feature that lets you do this automatically. Traditionally this capability is indicated by a tilde (~) on the board.

2 **RGB LEDs** LEDs are one colour, but you can get components with multiple LEDs built into the same shell. There are a few different types, but RGB ones are the most common. By using PWM, you can adjust the brightness of the different LEDs to make almost any colour.

3 **NeoPixels** (also known as WS2812B). These are RGB LEDs with a built-in controller. There's one input pin and one output pin. You can connect the output pin of one to the input pin of the next to chain these into long displays driven off a single microcontroller I/O. The only limits are the RAM in your controller, and the speed you need to update the LEDs.

4 **DotStars** (also known as APA102s). These are similar to NeoPixels, except that they take two inputs (data and clock). The update speed is much faster; however, they are also more expensive.

5 **FastLED** This Arduino library can control many addressable LED types (including NeoPixels and DotStars), and it also includes some very clever maths to help you manipulate colours quickly, even on slower microcontrollers.

6 **Animations Library** **hsmag.cc/8nZqaJ**. There's no need to reinvent the wheel, and many animations have already been written before. This library includes some ready-to-go animations to add a bit more life to your projects.

7 **On-board LED** Many Arduino boards have a built-in LED. Check the documentation for which pin it's on, or use the LED_BUILTIN constant. It's useful if you just need a status light. You can even flash in different patterns to indicate different error codes.



Below Table tennis balls are a great diffusion option



Right LEDs can look great when shone through a diffusion material, such as this canvas

7 SOURCES OF ELECTRONS

HOW TO GET POWER INTO YOUR PROJECT

1 9V battery Arduino Uno boards need at least 7V from a battery to work, so they are usually run with 9V batteries using a simple connector-to-power-jack lead.

2 LiPo battery These rechargeable batteries can provide quite a bit of power. If your board has on-board charging, then they can be really easy to use as well. However, they are a little delicate and can be dangerous if damaged, so make sure you understand the risks, and protect them properly if using them in a project.

3 Coin cells There are various types, and the model names usually start with CR, such as CR2032. The numbers are the diameter and height – in this case, 20mm wide and 3.2mm high. They're 3V, so you may need more than one to provide enough voltage for your project.

4 Mains If you're close to a power socket, mains electricity is usually the easiest option. USB power adapters are now ubiquitous, and many of us have more of these than we know what to do with. Some Arduinos can also be powered with barrel jacks, but be careful as these come in many different voltages – just because a power adapter fits, that doesn't mean that it's OK to use.

5 USB battery pack These aren't usually a very efficient way of powering Arduino boards because the batteries themselves run at around 4V. This is then shifted up to 5V for USB, then most boards run at 3.3V and usually throw away some of the power to get down to this voltage. However, the best component for a task is often the one you have, so if you happen to have a USB battery pack, they can work.

6 Solar The sun shines on most of the Earth for at least some time every day. You can use it to either power your project directly, or to charge batteries. Small solar panels can be quite affordable, so as long as you're careful with power usage, a solar-powered Arduino needn't cost too much.

7 Low power Most microcontrollers have some features to help minimise power usage, and if you take advantage of these, you may find that you can reduce the power consumption considerably. There's a library for SAMD21-based boards at **hsmag.cc/oBWyBV**, and an option for ATmega controllers at **hsmag.cc/8wEc41**. For other controllers, you will need to search online for further information. →

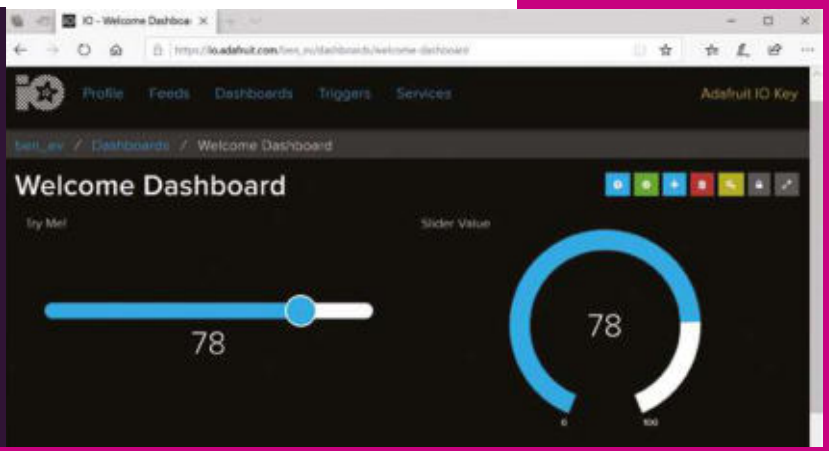


Left Coin cell battery packs – such as this one from Adafruit – are small and great for projects that don't demand lots of current

8 WAYS OF BUILDING AN INTERNET OF THINGS

MAKE YOUR OWN CONNECTED WORLD

Many Arduino-compatible boards can connect to the internet in some way, but in order to make them useful, we need something for them to communicate with. Fortunately, there's a wide range of tools for you to use to make this easy. Here, we look through our eight favourite IoT back-ends that we can use to make our online Arduino projects really useful.



Below
The Adafruit IO framework is great for bringing in sensor data. You can even trigger off-the-shelf hardware using IFTTT (If This Then That)

1 Adafruit IO is an online service that allows you to store bits of data that come from, or should go to, your internet-connected devices. It's based around the concept of streams of data. Each stream is a sequence of bits of information that could be numbers, text, or other things. Each is associated with the time it's added, so you can, for example, log the temperature as a stream of numbers representing the degrees Fahrenheit. On top of this, Adafruit IO has dashboards which each include widgets that can either read data from a stream and displaying (such as a line graph or gauge), or send data to a stream, such as a button you can click on and off.

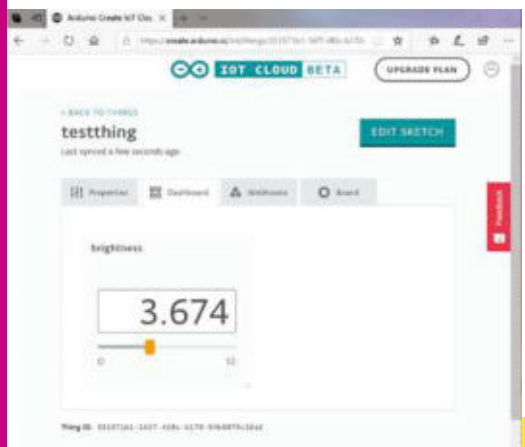
There are libraries for Arduino and CircuitPython to help you send and receive data.

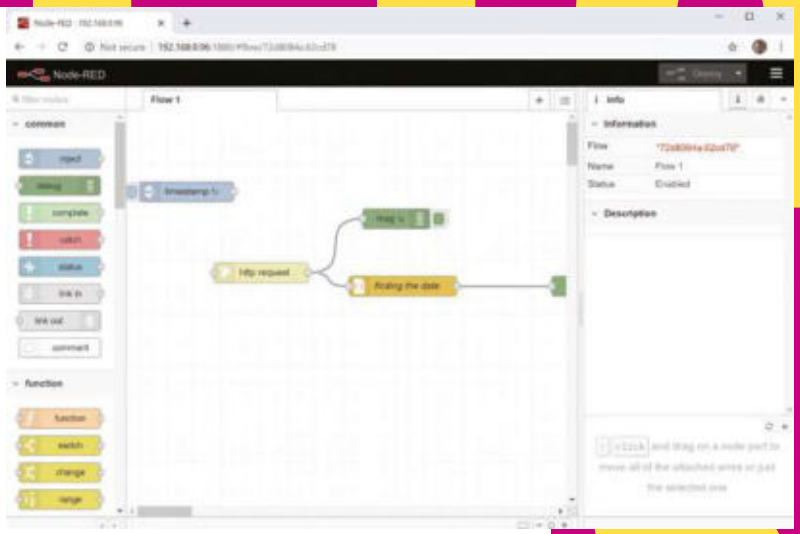
The free plan lets you create ten feeds and five dashboards with data stored for 30 days, but if you upgrade to pro (for $10 per month), you get unlimited feeds and dashboards with data stored for 60 days.

2 MQTT is a lightweight messaging system that's organised around topics. Devices can send data to topics or subscribe to them to receive data sent to that topic. It's the underlying technology of Adafruit IO. Servers are known as brokers, and there are some publicly available ones that you can use (though they usually don't offer much in the way of security or guaranteed uptime). However, the server software is open source, and you can run it on your own system. This means that you can keep the data entirely on a local network, so it's not dependent on the internet and can be locked down for increased security.

3 Node-RED This flow-based programming language is great for simple IoT projects. Essentially, it allows you to link up a wide variety of inputs with a wide variety of outputs and has options for manipulating data along the way. It all runs in the web browser, and you can drag-and-drop different nodes and link them together in various ways.

It's best suited to event-based systems where an action is triggered by a certain event. Depending on your needs, this might mean that it's best deployed alongside some other system.





4 Particle.io is not strictly speaking Arduino as it doesn't use the Arduino IDE, but this IoT system uses a series of custom boards and a web-based IDE, and is programmed in a language almost identical to Arduino. As well as gathering data from deployed devices, it can manage a fleet of deployed devices to help keep your own network of things running smoothly.

5 The Arduino IoT Cloud makes it very easy to build web-controlled devices. Its web-based IDE lets you link properties between an online dashboard and your physical hardware. This goes both ways so you can have buttons on the web dashboard that control the behaviour of the device and also information from the device can feed back into the dashboard.

6 The Blynk system is free for non-commercial use, and lets you create a phone app to control your hardware. It's perhaps more suited to creating independent things rather than interconnected systems (such as smart homes). While there is specific Blynk hardware, it also works on a wide range of Arduino-compatible internet-connected hardware.

7 Home Assistant is a general home automation framework and, in most cases, it's used with off-the-shelf IoT products. However, it also has a module (integration in Home Assistant speak) for working with Arduino. Arduino control is a bit rudimentary, but it allows you to switch pins on and off and read sensors.

8 openHAB Similar to Home Assistant, openHAB is mostly used with off-the-shelf products. However, it can be expanded with DIY Arduino-based devices through the MQTT interface. This is perhaps slightly more complex than adding devices to Home Assistant, but it gives you much more flexibility in how you use your hardware. →

Above
The flow-based nature of Node-RED makes it easy to structure your projects

Left
The Arduino IoT Cloud creates dashboards for controlling your creations

8 CODING TRICKS

EXTRA FEATURES OF THE LANGUAGE YOU MIGHT NOT KNOW

1 Macros You might think that Arduino is one language, but actually, it's two! There's a preprocessor that runs before compilation, and you can give it commands to manipulate your code. The most common of these is `#include`, which copies and pastes code into your sketch (usually libraries), but there's also `#define` which is a bit like 'find and replace'. It will search your code for a bit of text and replace it with another.

2 Map function This takes a number that's a proportion and converts it to another scale. For example, if you read in an analogue value, it comes in as a number between 0 and 1024 (or other depending on the resolution of your ADC), but this might really represent a distance between 0 and 1 m, or a temperature, or an angle, or anything else. The `map` function converts between these different scales.

3 Delaymicros() You've probably used the `delay()` function to pause your program for a number of milliseconds, but if you need a really short pause of less than one millisecond, you can use `delaymicros()` which does the same as delay, but in microseconds.

4 PulseIn This slightly archaic function waits for a pin to go either low or high (configurable in the parameters), then times how long it stays in that state for and returns this time. This function can be useful for lots of things, such as creating an interface where long and short button-presses have different meanings.

5 Timers Getting things to happen when you want can be a complex problem on Arduinos. Depending on your hardware, there should be two or more hardware times that you can access. They're incredibly flexible, but can be a little awkward to use. If you don't need complete control of the timers, there are some libraries that make matters a little easier, such as Timer (see **hsmag.cc/4JTuHU** for details).

6 Interrupts These let you specify what code you want to run when a particular event happens. For example, when a pin changes state, run this function. They let you create powerful event-based code very easily. See HackSpace magazine issue 12 for more details.

7 Port manipulation You may be used to using `digitalwrite()` to set the values of pins. This works well, but includes a load of checks to make sure you're doing something sensible, and these checks take time. You can set the output directly by manipulating the ports – see the documentation at this link for more details: **hsmag.cc/sGMrV8**.

8 Compartmentalise You might be used to a sketch being just a single file, but almost all are multiple files that are brought together with include statements. You can make your code more manageable by splitting it up into bits and, ideally, building your own libraries for things you use regularly. You can find a handy tutorial on how to do this in the Arduino documentation at: **hsmag.cc/tTUtRJ**.

8 WAYS TO PROTECT YOUR PROJECT

DESIGN AND MAKE YOUR IDEAL ENCLOSURE



Above
Extruded aluminium cases, such as this one from Adafruit, are tough and affordable

1 If you want a box for your Arduino made quickly, then a laser cutter is the best option. Using plywood or acrylic, you can get a box made in a custom shape in a few minutes. Using a tool like **en.makercase.com**, you can enter the dimensions you want, and it will output a laser-cuttable design.

2 3D printers are another way of quickly building enclosures for your project. You can take a look at the Thingiverse tag project_box for inspiration (**hsmag.cc/qfJNe3**), or you can just design one to your own requirements.

3 Tupperware is perhaps the cheapest off-the-shelf option. It can work well, but it's a bit tricky to work with as it's prone to cracking. You can melt holes in it, but be careful of fumes.

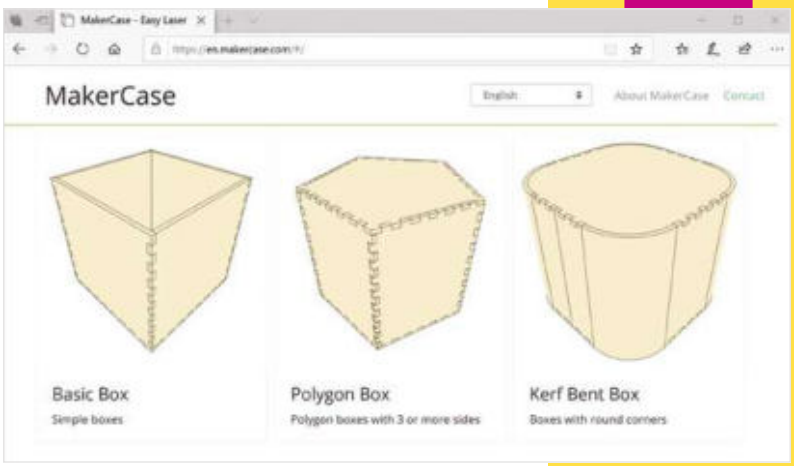
4 Extruded aluminium cases are very solid if your project is going to be put to rugged use. Just be aware that aluminium is a conductor, so you need to make sure your project is properly mounted inside the box. Many aluminium cases include slots for holding some sizes of PCB in place, but check that they're the right size for your project.

5 Many craft shops sell cheap plywood boxes that make excellent project cases. They're not usually very strong, but they can look good, especially if varnished or painted.

6 There is a huge range of commercially available plastic boxes for projects. Many of them have some degree of waterproofing, and this is defined by their Immersion Protection (IP) rating. IP54 is resistant to dust and splashed water, so is a good choice for indoor projects.

7 IP67 is dust-tight and able to withstand immersion in water up to 1 m deep (the test period for this is 30 minutes). This is suitable for outdoor projects in sheltered locations. It should withstand most rain and the occasional dunk, but not necessarily a complete and utter soaking.

8 IP68 is a bit more of a loose standard. It should be able to withstand immersion deeper and longer than IP67, but the exact test varies from manufacturer to manufacturer, so they should be able to provide details on exactly what. →



Above
Websites such as MakerCase let you quickly and easily design a box for laser cutting

TOP 5 ARDUINO PROJECTS FROM HACKADAY

GET INSPIRATION FOR YOUR NEXT PROJECT

There are loads of places that people can upload their projects to. These websites serve as inspiration for others, a guide for how to do things, and also – particularly in the case of **Hackaday.io** – a collaboration platform, where multiple people can work together on a single project. It's worth taking a look around **Hackster.io**, **Instructables.com**, and the Arduino projects hub (**hsmag.cc/NuS54T**) to see the huge range of projects that you can complete with these microcontrollers. Here, as an example, are the top five projects from **Hackaday.io** that are tagged with Arduino.

1 3D-printable robotic arm The 'Mantis' robotic arm took part in the 2015 Hackaday prize, and has continued development since then. This is designed to be able to lift 2 kg and move with greater precision than small, cheap 'servo' arms. 3D print files are available, but build instructions are a little sparse.

2 Arduino Glasses These use an OLED display, a mirror, and some acrylic to provide a 'heads up' display that clips on to glasses. The original intention is to use them to display multimeter readings so you can see the values while keeping your eyes on the circuit you're testing.

3 PCB mill for under $10 Creating PCBs quickly and easily is a big challenge for many electronics enthusiasts. You can order them cheaply, but the lead time is often over a week, which is a long time to wait only to find out there's an error. You can use strange chemicals and pens, but it's fiddly at best and dangerous at worst. Milling a PCB is perhaps the best option.

4 MIPI DSI Display Shield/HDMI Adapter Mobile phones have high-definition displays, and these are available as spare parts for relatively little money. Unfortunately, it's not particularly easy to use these with other hardware. This project got an iPhone 4 screen running with an Arduino shield (using an FPGA to do the heavy lifting). While the project hasn't been updated in several years, the hardware design and firmware are available for anyone looking to pick it up.

5 Tiny Robot Family The ATtiny series of microcontrollers is probably the least powerful of all Arduino-compatible microcontrollers, but they're cheap, small, and through-hole, so are easy to work with. This project creates a series of different robots that move according to different algorithms, and therefore display different behaviours.



Right ◈ **Browse through some great Arduino projects at hsmag.cc/J7eVRq**

5 EXAMPLE PROJECTS

LEARN ABOUT ARDUINO BY DIGGING INTO THE EXAMPLES

You've probably seen that the Arduino IDE comes with a set of example sketches to help you get started. However, you might not have seen that there's a website with more details of these examples, including guides on how to wire them up. You can browse through all of them at **hsmag.cc/ohjw28**, but our five favourites are:

1 Tone Keyboard
The easy way to make noise

There are many ways of making sounds on the Arduino, and the tone library is probably the simplest. You don't get many features, but you do get enough for simple tunes or audio cues to help users.

2 AP Web Server
Create a WiFi network

Different WiFi-connected devices handle WiFi in different ways, but this example is for the official Arduino web-connected boards. It starts a wireless network and creates a web server that lets you turn an LED on and off. While this is basic, it's a great starting point for your connected projects.

3 Read Write
Storing data between sessions

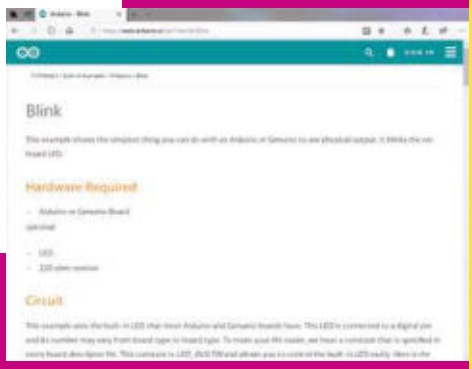
SD cards are ubiquitous, cheap, and easy to interface with microcontrollers. This sketch takes you through reading and writing data to files on an SD card. These can also be read or written to from a computer, giving you a way of transferring large quantities of data (such as historical sensor readings) on or off your Arduino.

4 Firmata
Computer control

This protocol lets you send commands over a serial connection between a computer and an Arduino. There are a few ways of using it, from letting your computer take complete control of the I/O pins on the microcontroller, to sending more advanced commands such as directions for robotic movement. There is also a range of libraries for different programming languages that can run on the computer side to interface with this.

5 Character Analysis
How to manipulate letters

Strings can be a tricky data type to get your head around, but there are quite a few operators to make it easier for you. This example (and others in this section) guide you through the various text operations available to you in the Arduino programming language. ◻



Below
The official examples come with code, circuit diagrams, and explanations – they're a great starting point for gaining a deeper understanding of the platform

SUBSCRIBE TODAY FROM ONLY £5



SAVE UP TO 35%

Subscribe today and get:

- **FREE delivery**
 Get it fast and for FREE
- **Exclusive offers**
 Great gifts, offers, and discounts
- **Great savings**
 Save up to 35% compared to stores

Subscribe online: hsmag.cc/subscribe

SUBSCRIBE TODAY

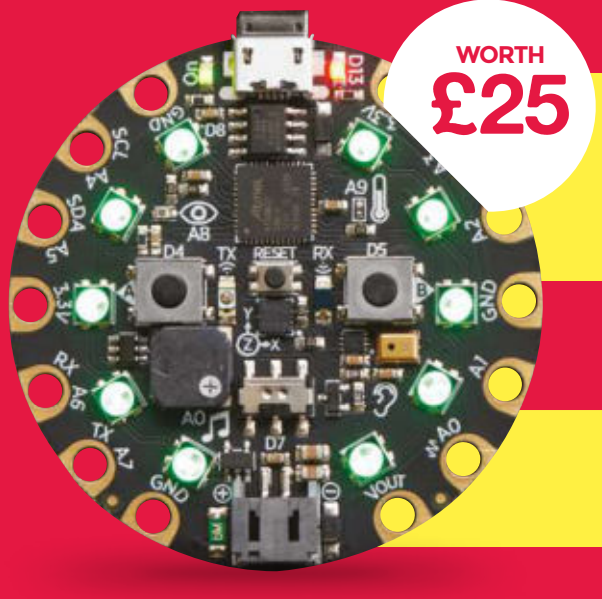
Subscribe for 12 months | **Rolling monthly subscription**

£55 (UK) **£90** (USA)

£80 (EU) **£95** (Rest of World)

Free Circuit Playground Express with 12-month upfront subscription only (no Circuit Playground Express with rolling monthly subscription)

- **Low monthly cost** (from £5)
- **Cancel at any time**
- **Free delivery to your door**
- **Available worldwide**



FREE!

Adafruit Circuit Playground Express

With your 12-month print subscription

This is a limited offer. Offer subject to change or withdrawal at any time.

SUBSCRIBE on app stores

From £2.29

Available on the App Store

GET IT ON Google Play

Buy now: hsmag.cc/subscribe



FEATURE

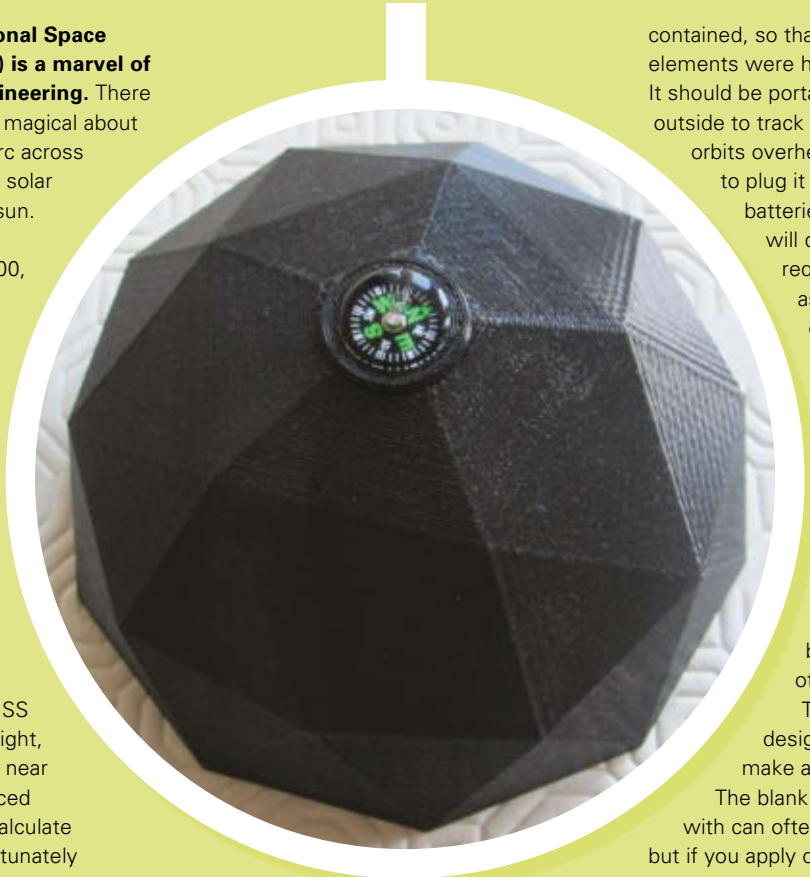
How I Made
ISS TRACKER

A portable guide to the night sky

By **David Hunt**

The International Space Station (ISS) is a marvel of modern engineering. There is something magical about watching it arc across the sky as its solar array is illuminated by the sun. Since the first crew took residence in November 2000, the ISS has been orbiting the Earth every 90 minutes at a speed of five miles per second! NASA shares a range of live data about its position which enables earthbound observers to track its orbit.

The ISS orbits at an angle of 51.6°, which means it can cover 90% of the planet's surface as the Earth rotates below it. The best time to view the ISS is when it is reflecting sunlight, and this most often occurs near sunrise and sunset. Advanced mathematics are used to calculate the orbit of the ISS, but fortunately this information is readily available as live data if you have an internet connection.



Above I replaced the compass with the stars

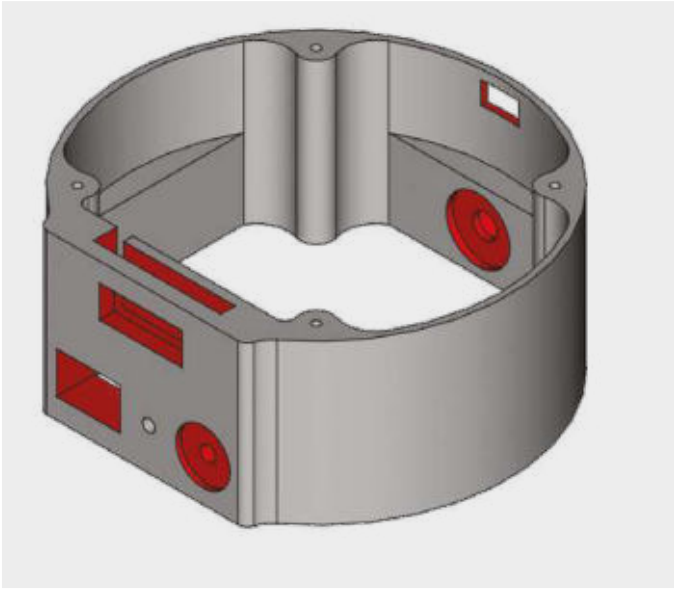
DESIGN PHILOSOPHY

When designing the concept for this ISS Tracker, I wanted the unit to be self-contained, so that all the mechanical elements were housed in an enclosed unit. It should be portable so it can be taken outside to track the flight of the ISS as it orbits overhead, but it would be good to plug it in to recharge the batteries as the stepper motors will quickly use up power. This requires an ingenious design as the unit must be able to continuously rotate without tangling any wires. It also needs to be internet-enabled to access a website to download the current ISS location data. The final consideration was that the design should be open-source, and the component parts should be readily available for other makers to replicate it.

There were some key design decisions I had to make at the start of the project. The blank piece of paper you start with can often be a bit intimidating, but if you apply one of the principles of computational thinking and decompose the problem, it is possible to break down the big problem into smaller and more manageable pieces. I initially considered using a construction kit, such as in Grady Hillhouse's excellent solution. This has

the advantage of a quick build time, but I decided to design a bespoke solution that could be 3D-printed, as this provided creative freedom with the design features and was not limited by a small range of components. I wanted the unit to have a small footprint, and modelling the design using FreeCAD enabled me to minimise this.

Right
3D printing makes it so much faster to prototype parts



COMPONENTS

28BYJ-48 stepper motors were chosen to rotate the device body and pointer, as it was easy to convert a calculated angle into a number of steps to arrive at the correct pointing position. Servos were an option, but the pointer would have to 'sweep back' once every orbit in order to keep track of its position, which I considered a less elegant solution.

The unit is powered by two 18650 Li-ion cells, recovered from a laptop battery pack. These cells need to be handled with care as they can be damaged, or even explode, if overcharged. A 6 V power adapter can be plugged into the rear, which connects to a battery charger module. This is essential to balance the charging current to each cell to prevent overcharging. The cells are in series, providing a nominal voltage of 7.4 V, reduced to 5 V through a switched buck converter, which is used to power all the motors and electronics.

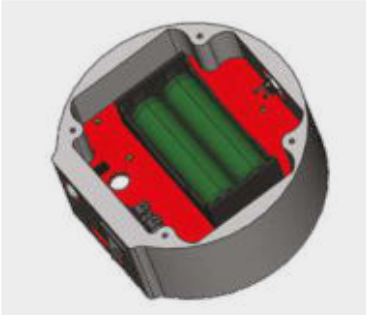
At the front of the device, there is an OLED display. These do not use a backlight and can provide high luminosity displays with sharp text – this is ideal for this application. The display resolution is 128 by 32 pixels, and it can provide two lines of text clearly to give the user instructions and information about the ISS location.

> The best time to view the ISS is when it is reflecting sunlight, and this most often occurs near sunrise and sunset

Below
The base of my design now contains almost all the electronics



Below
Laptop batteries can be useful, but you have to be careful with them

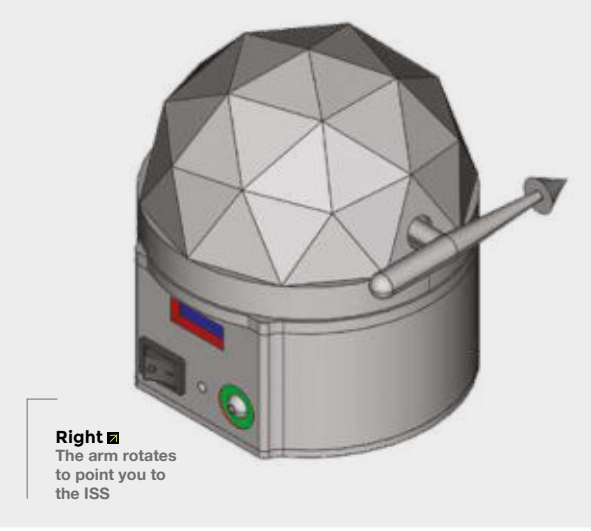


The stepper motors are driven using ULN2003 driver boards. These are great circuits to use when prototyping new designs as they have a row of LEDs that show when the coils of the stepper are energised, so not only can you tell if the motor is being driven, you can also estimate how fast it should be moving by the speed the lights flash.

The rotating platform is secured in place with a lock nut that houses a six-channel slip ring. This allows the power and drive signals to come from the lower stationary base of the device while the top section is spinning. This is a key feature of the design, and the slip ring enables it to continuously rotate without the connecting wires getting tangled. The rotating platform has a ball bearing embedded in the centre of the disc to let the top section spin with very low friction.

The ESP32 microcontroller runs on 3.3 V, but there is a voltage regulator on the development board that handles this. A micro USB connector can be used to connect to the ESP32 without dismantling the device, as we all know that designs are never entirely complete – there is always more tinkering to be done! The average current of 450 mA used by the device indicates it could run on these cells for over six hours. This is ample time to set it up to view the fly-by. If it is to be run continuously, it will require mains power to meet the demands of the stepper motors, which use electric current even when the motor is not rotating.

FEATURE



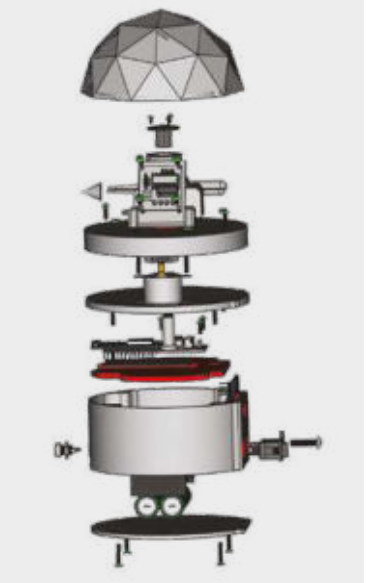
Right
The arm rotates to point you to the ISS



Left
Inside the case, there's a lot going on

The ESP32 microcontroller has a built-in WiFi antenna, so it can readily connect to a router and download data remotely while viewing the event outside. When using switches as input devices, the designer has to consider the problem that occurs when the contacts of the switch bounce against each other. This will be seen as a series of on-off inputs by the microcontroller. I solved this common problem using a built-in touch sensor as a switch. This means it does not need to be 'debounced'. The single switch is used to calibrate the starting position of the pointer as the device goes through its startup routine, so that it has a reference point from which to measure the required angles to locate the ISS.

Below
Digital design gives you great images!



DESIGN DEAD ENDS

Although it is important to think about the features of your project, there will always be opportunities to improve the design, which you will only realise once you are in the process of making it (however experienced you are). These discoveries can sometimes be frustrating as you kick yourself for not thinking of them earlier, but this process of developmental improvement should be celebrated – you have learned something new along the way!

This process of developmental improvement should be celebrated – you have learned something new along the way!

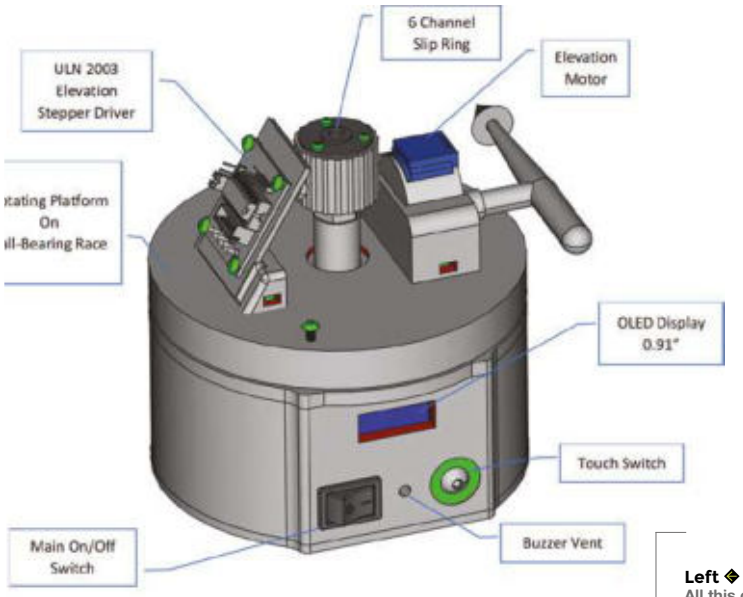
As I required a rotating platform, I thought I might use a slewing bearing (which is a mini project in itself). The advantage of using this component is that I could design it to the dimensions I wanted, and the internal hole could be made big enough to allow the wires to pass through it. I successfully made the slewing bearing, and it worked – but, to my disappointment, the rolling action of the bearings was poor as the surface finish of the 3D-printed components was too crude. I eventually opted for a mass-produced roller bearing, commonly found in skateboards, which had a beautifully smooth action.

I knew that the user had to point the arrow to north when calibrating the device; I thought the easiest way to do this was to add a traditional compass. I used Blu Tack to temporarily stick a button compass to the top of the dome to check that it would work without being influenced by the magnetic fields of the stepper motors. It seemed to be far enough away at the top of the dome to give an accurate reading. It was only when I was testing the device that I thought about using the pole star (Polaris) as a reference point. As I would only be searching for the ISS on clear nights, it should not be a problem to use the stars for navigation, so I removed the compass and retained my mathematically perfect geodesic dome as a top cover.

The biggest design dead-end I encountered was when I had to rethink the

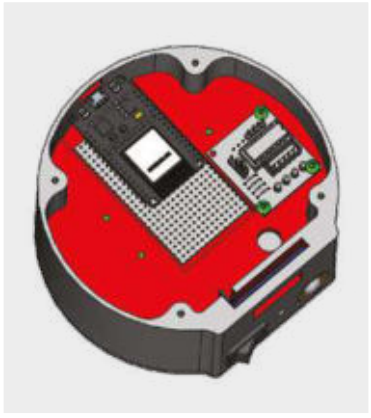
'layers' of the design. I initially started with the battery in the stationary compartment at the bottom and the ESP32 on the rotating platform at the top. This worked well until I tried to add the display. I could use the wires in the slip ring to connect the microcontroller to the OLED display, and it worked well when stationary. As soon as the platform started to rotate, I found the slip ring was electrically noisy, and the 100kHz information was not transmitted accurately, resulting in a garbled message on the display. Back to the drawing board! I decided to keep the battery, microcontroller, and display in the non-rotating section at the bottom so that it was easier to connect them together. The elevation motor would be the only component that rotated in the top section, and this could be controlled with signals that changed at a much lower frequency, so could be passed through the slip ring without error. The benefit of this design change meant that the unit was now much more compact.

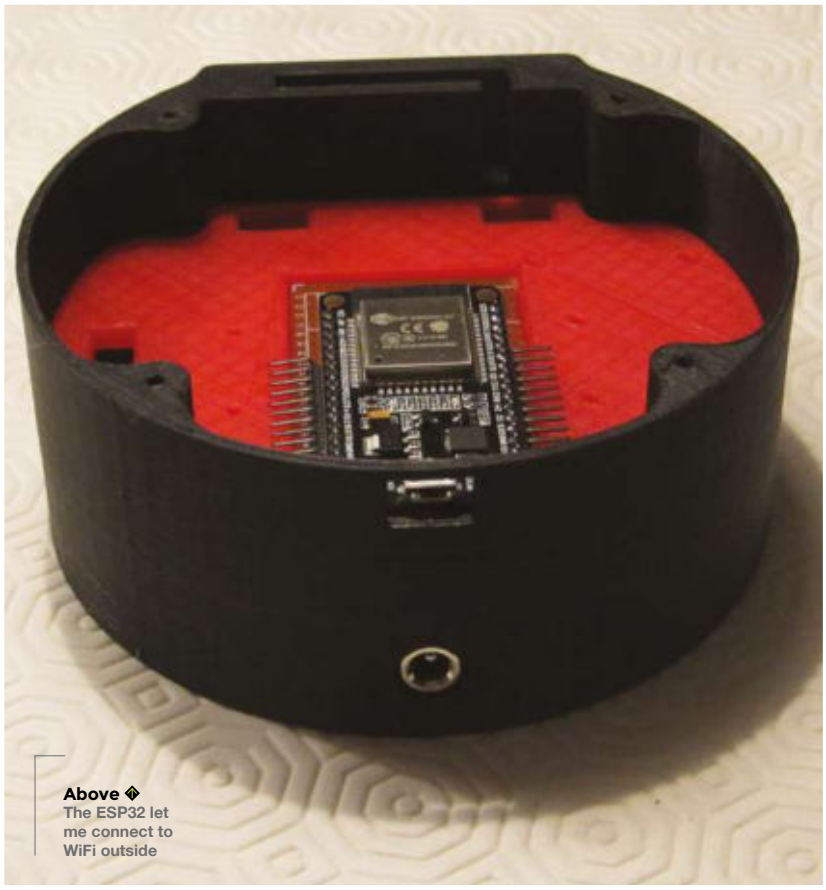
Of course, you always try to think ahead to avoid going down these design dead-ends, but I would argue that often it is worth taking the risk and trying to solve these problems in different ways. It might work, or you might learn something. As a maker, I usually adopt the 'Ready, Fire, Aim' approach as this allows you to get started, and you can refine your ideas as your project starts to take shape.



Left
All this complexity is hidden from the end-user

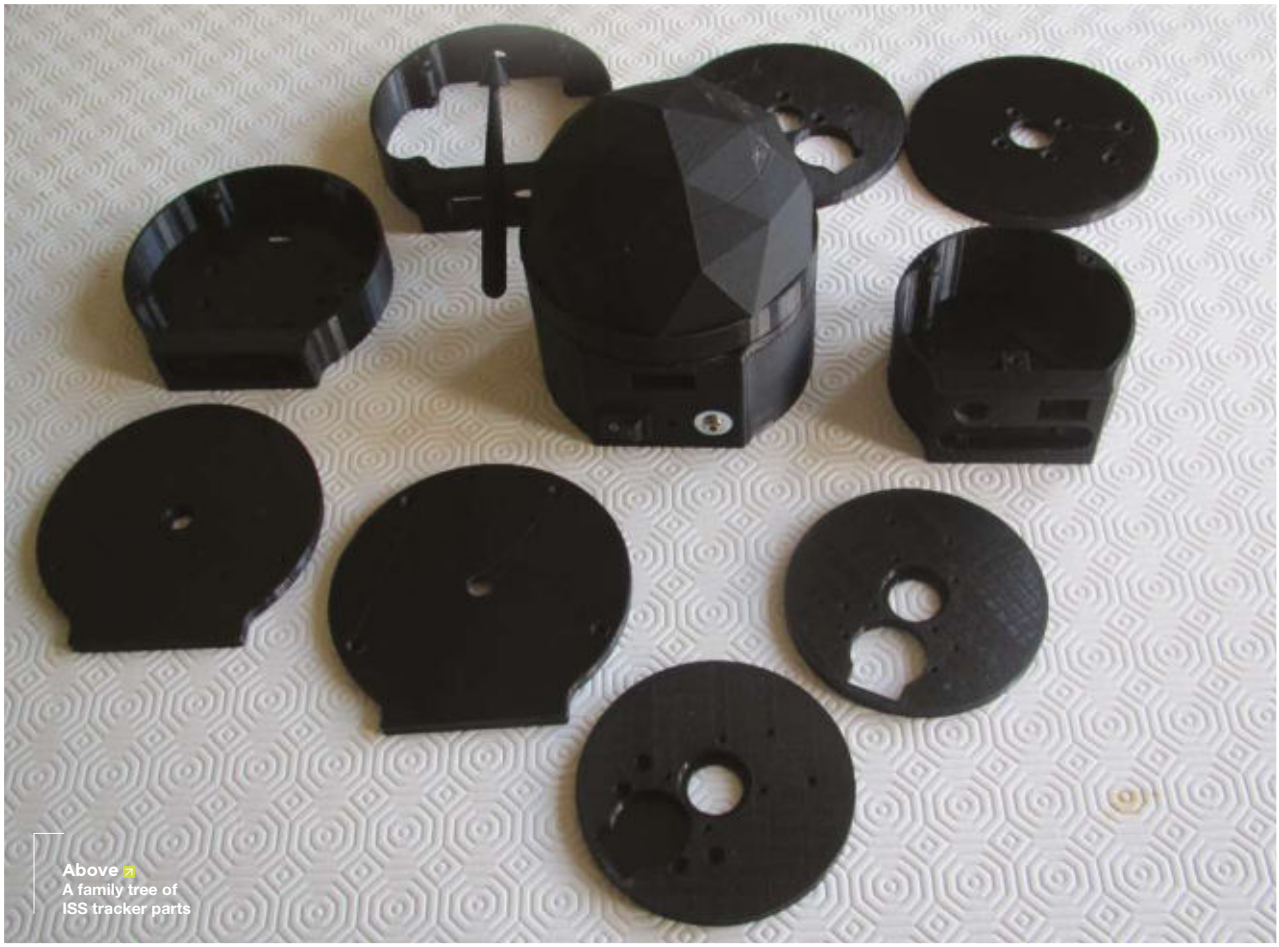
Below
The small microcontroller fitted snugly in the base





Above
The ESP32 let me connect to WiFi outside

FEATURE



Above
A family tree of ISS tracker parts

HOW IT WORKS

The ESP32 connects to the internet and downloads the position of the ISS, presented as a set of latitude and longitude co-ordinates. At any one moment in time, you can plot a point on the Earth's surface directly below the ISS, known as the nadir.

> At any one moment in time, you can plot a point on the Earth's surface directly below the ISS, known as the nadir

If you project a line straight up (or normal to the sphere of the Earth), the line will pass straight through the ISS. The co-ordinates of that point on the Earth are the ones that are to be found in the downloaded file. The file is in JSON format, which is human readable, but the important latitude and longitude values still need to be extracted from it.

The Open Notify website provides an API which allows you to download current ISS position data in JSON format. This text-based data can be read by a microcontroller.

This JSON file needs to be parsed to extract the position information, so that the text in the file can be used in the calculations to work out the pointing angles. If you know your own location on Earth (O) as well as the position that the ISS is directly over (N), it is possible to calculate the straight-line distance between these two points. The distance that the ISS is orbiting (h) and the radius of the Earth (RE) are known quantities. These measurements provide enough information to apply some trigonometry and calculate the angle to the location of the ISS from the observer's viewpoint.

If we can calculate the angle α (alpha), it is possible to elevate the device pointer to the correct position. If the pointer is initially pointing straight down towards the centre of the Earth, it needs to rotate through α degrees in order to point at the ISS.

The ISS position is constantly being monitored, so when the orbital distance decays due to atmospheric drag (there is a very slight atmosphere at this height), it is boosted to maintain its height. In 2019, this ranged from 405 km to 418 km, but to keep the calculations simple, an average orbital distance of 416 metres has been used.



Right
The final build tracking the ISS



Above
Checking we're at the right latitude

OPERATING DETAILS

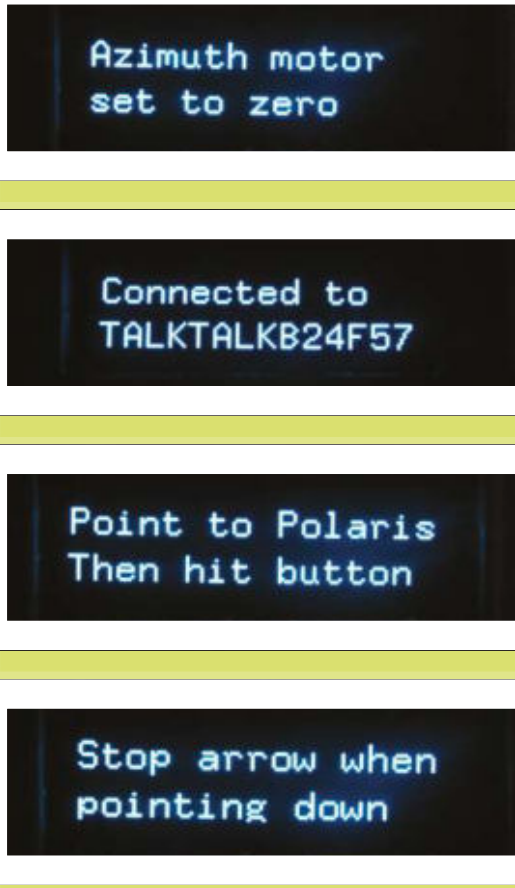
When pointing at an object in the sky, astronomers use two basic measurements to point their telescopes, the azimuth (left-right rotation) and elevation (up-down rotation). Once the correct angles have been calculated, the stepper motors need to move to these positions, but they need to have a reference point from which to measure.

When the unit starts up, the zero reference point for the elevation is straight down (towards the centre of the Earth) and then the reference for the azimuth needs to point to true north. To help with this, the pointer elevates to the same angle of latitude as the observer above the horizon. This means that the pointer should be pointing at the pole star. The fact that the height of the pole star above the horizon is equal to the observer's latitude was useful for ancient mariners to locate their position around the globe – finding the longitude was a lot harder as they needed to keep accurate time at sea and the motion of the waves often upset their delicate clock mechanisms (Sobel, 2005). Once the calibration has taken place, the device will continuously download the ISS location (every five seconds) and point the arrow towards it.

As the ISS orbits the Earth every 90 minutes, the pointer will move slower than the hour hand of a clock. However, if you watch it for a few moments, you will see that its position creeps along slowly, informing you that something is happening inside the device.

Once the basic mechanics of the device were working, it was time to write the code. The sophistication of the software will determine the tracker's abilities. Currently, it will track the location of the ISS and alert the user when the next fly-by is due. Potential developments include calculating if the fly-by is high enough above the horizon to view easily or scraping some local weather data off a website to assess if it would be visible. As mentioned earlier… are projects ever truly complete?

I am left with one philosophical question, similar to that tree in the forest; if no-one sees the ISS flyby… was it ever there? ■



Above
The OLED display is easy to read, even outside

Further reading

Websites to track the ISS
hsmag.cc/ISStracker
hsmag.cc/SpotTheStation
hsmag.cc/SatFlare

Height of the ISS
hsmag.cc/HeavensAbove

ISS Current Location
hsmag.cc/ISSLocationNow

Sobel, D. (2005) Longitude. London: Harper.

IN THE WORKSHOP: Music to our ears

By **Ben Everard**

Adding internet streaming to an old radio

Several years ago, I came into possession of an old radio which didn't work. While I'd love the challenge of restoring it into a working radio, the truth is I simply don't have the time to learn enough about valves to even know where to start about getting it working. However, the speaker still worked well, and it looked lovely, so I replaced the innards with a KRC-86B Bluetooth receiver and an amplifier. The Bluetooth receiver worked well for several years, but stopped working this month. Faced with no music in the kitchen, I initially thought of replacing it with another Bluetooth receiver, but there are a couple of problems with the KRC-86B that I wanted to address.

1) There's no way of changing the display name from KRC-86B to something more descriptive. This is a problem if you have more than one of them.

2) The range of Bluetooth is alright, but not ideal. As long as the phone was in the kitchen, this speaker worked fine, but if it left the room, the music started stuttering and skipping.

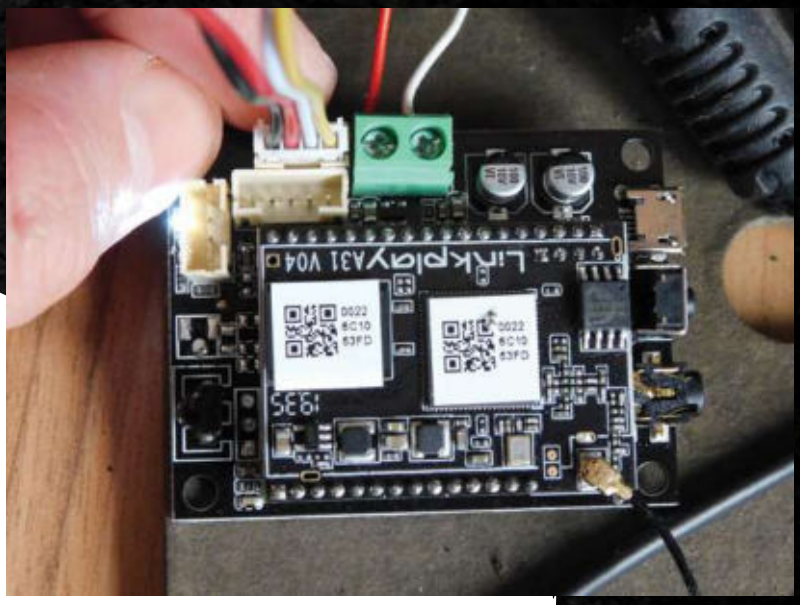
The solution to both of these problems seemed to be a WiFi speaker. The Sonos brand is the most famous of these, but there are others. With more processing on board, it's usually possible to change the name of these, and they're not tied to the Bluetooth range. There are a few routes you can go down with these – buy an off-the-shelf speaker and pull out the innards, use a Raspberry Pi with appropriate software, or use a bare module designed for the purpose. We opted for the latter option and got a Up2Stream Mini V2 delivered to our door for £23.10.

Setting up this device was really simple. Install the 4STREAM app, then power on the device (via a micro USB connector or 5V screw terminal). It'll create its own WiFi network which you can connect to. Open the app, and it'll guide you through setting



Below
The radio up and running in my kitchen

It's a definite upgrade on the old Bluetooth setup



it up, which includes selecting the network you want it to connect to. Once this is done, restart the device, and it'll connect to your home network. In the app, you can then select the music you want to play.

You can play MP3 files directly from your phone, but ours struggled a bit with this, and I got occasional pauses in the music stream. This might just be due to my phone. The main reason we got this device, though, was to play music directly from Spotify (the device can also stream from Tidal, Napster, Deezer, and Pandora) – and it worked flawlessly.

Once everything is set up, the Up2Stream is recognised by Spotify as a device, and you can select it in the Spotify app.

The Up2Stream Mini V2 outputs low-power audio, so it needs an amplifier if you want to power more than just headphones. We had an existing amplifier board that we had been using with the Bluetooth setup, so this just slotted into place. If you're starting from scratch, there's an option that comes with a built-in amplifier – we might try that for our next speaker setup.

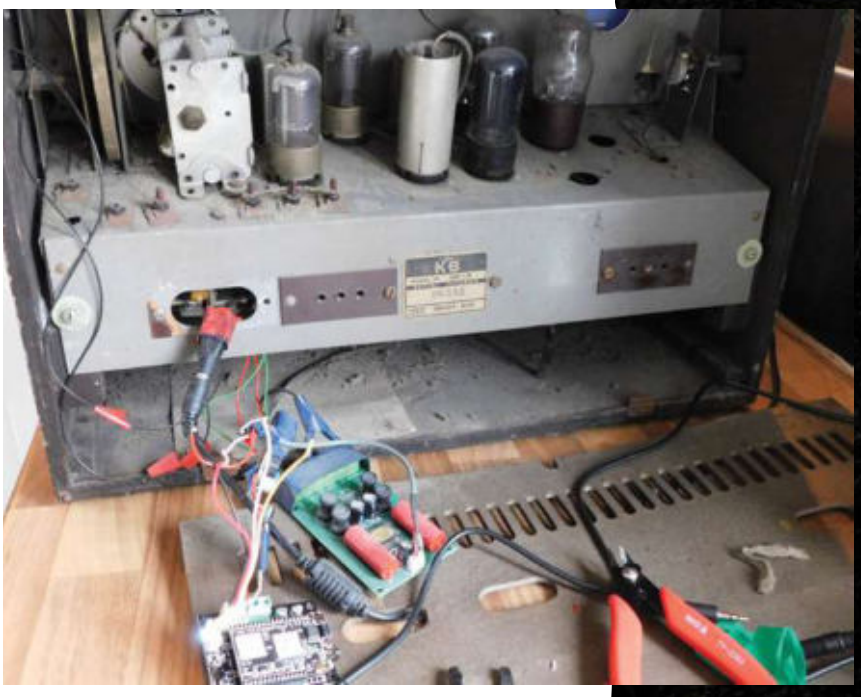
DAY TO DAY

I've been using this speaker for a few weeks now, and I'm pretty happy with it. The only problem is a very slight buzz when there's no audio being played. It's only noticeable when the room is completely quiet, and it's easily solved by just powering off the radio. The other downside compared to Bluetooth is that it doesn't just forward *all* audio to the speaker, so for example, you can't play YouTube audio or BBC Sounds podcasts (at least not via Android – it might be possible to do this with AirPlay on iOS devices, but I don't have one to test).

I got this device as an experiment with few expectations of how it would work, and I've been really impressed. It's a definite upgrade on the old Bluetooth setup, and it's let us keep this old radio in regular use. We think it's coming up to 80 years old now, and we're still using the original speaker. There are thousands of other old radios, hi-fis, and speakers out there not being used. Devices like this make it much easier to upgrade them to the internet age. □

Above
In its simplest form, you can power via micro USB and take audio via the 3.5mm jack

Below
Yes, at some point I will dust the inside of the radio, and sort out proper mounts for the boards



HackSpace magazine meets...

Svetlana Quindt

Cosplay, creativity, and the art of getting things wrong before you can get things right

Whenever we make a thing, we're happy as long as it works. Function comes first, and form... well, form is all very well if you've got the time, but as soon as we've finished Thing A, we've moved on to Thing B with half an eye on Things C, D, and E.

There's one subset of makers who don't share this mindset: cosplayers. For a costume to be a success, every aspect of it has to work. Electronics, metal, foam, fabric, 3D-printed parts, all have to look right, as even the best work can be undone by shoddiness elsewhere. The range of skills cosplayers have to learn and implement, flawlessly, is something to behold.

To get an insight into the world of cosplay, we spoke to Svetlana Quindt. Her website, KamuiCosplay.com is a treasure-trove of creativity, guiding the viewer from first sewing pattern, to customised 3D modelling designs that the user can transfer to foam to make realistic armour, sewing patterns, STL files, wiring schematics, and more. →

Right
This costume, from World of Warcraft, was Svetlana's first stab at incorporating lights

HackSpace We've got to get this out of the way, Svetlana. We know next to nothing about any of this stuff. We can see the skill that goes into your work, but Dragon Ball Z, Monster Hunter, League of Legends… they're all unknown to us. What is it about video games that inspire people to make costumes?

Svetlana Quindt I think the fascination with video games is that, while you can read a comic or read a manga, or you watch a cartoon, you're always consuming. But when it comes to video games, you become the hero, so you have a much stronger connection to your character.

Video games also have a higher level of difficulty when it comes to making costumes, simply because they have more details: they have light effects, they have more cool props and stuff, and it's also easier to get proper references from the character. Sometimes with a comic or a manga, it might be difficult because you'll only have a tiny little picture. But with video game characters, you can see how they animate. It gives you more to work with.

HS How did you get started making cosplays?

SQ I got started back in 2003 when I was watching *Dragon Ball*. My very first cosplay was the Great Saiyaman from *Dragon Ball Z*. That was in 2003. I discovered the cosplay scene in Germany and kept on doing it, and it was there that I discovered Worbla. It's a foam-like material that's popular among cosplayers here. It's really easy to work with, flexible, and holds paint well, so you can get a good finish with it. I started playing World of Warcraft, and I fell in love with my own character – I play a Night Elf Druid, so of course I made the costume.

This was the point at which I got into electronics. The armours often light up in the game, and I wanted to recreate this effect, so this was the first time I started to experiment with LEDs and electronics and everything. I just kept on doing it. Over time, people kept asking me how I made this or that. Or this armour, how I made simple pieces of my costume. Since it was so difficult to keep answering again and again because I was getting so many messages, I started doing a YouTube channel. I started making tutorials on YouTube with my progress pictures and stuff like that. A few years later, people were asking me if I could put all these tutorials in one place because everything was spread across social media – YouTube, and everything.

A friend finally convinced me to write my very first book, which was about working with Worbla – the plastic modelling material I mentioned before. I wrote a book that taught people how to get into cosplay. For the last five years, my husband and I have been self-employed, writing 'making cosplay' crafting books. It's a long story, but I'm pretty happy with it.

> "I've travelled a lot in the past. The very first year I started doing this full-time, we had 18 conventions"

HS So, you've gone professional?

SQ I don't like to call myself a professional cosplayer, simply because I don't get money all of a sudden for doing what I've always done: I go to conventions, hang out with my friends, show off what I've been doing, and check out what other people are making. I make an income from teaching people how to make costumes, how to make props, how to install lights, how to sew, and all that. I really try to help the cosplay community.

HS There's obviously a growing scene for cosplay in Germany; I know it's a big thing in the USA too. Do you go around the world to conventions?

SQ I've travelled a lot in the past. The very first year I started doing this full-time, we had 18 conventions – I just wanted to see as much as possible.

Travelling became a bit stressful for me, partly because I prefer to be at home making things, writing books; partly because of all the jet lag and everything. Also, climate change. We decided that from this year we would stop going to conventions by plane, so we only take the car or public transport. I want to reduce my CO_2 footprint and become a little bit more eco-friendly.

I also want to take longer over projects, use them for the maximum lifetime and get plenty of tutorials out of them – not just make a costume in one week, move on to the next immediately and have a load of wasted materials.

HS Looking at your website, I can see you've produced books on painting, EVA foam, armour, props, sewing, makeup, electronics…

SQ I didn't write the makeup book – that's by a friend of mine. I actually have books from my friends in my shop as well. I want to show that it's totally OK to do.

We also had some books from Punished Props, from Bill Doran, and we try to collaborate and show the community that it's OK to do the same thing and still support each other.

HS I've seen a lot of that sort of spirit in the maker universe. There's a lot of people supporting one another. It's not a competition; people are in it for the fun.

SQ It is a competition, especially when it comes to cosplay contests!

In general, though, people tend to compare themselves with others; you see all these amazing, crazy costumes and you might think: "I will never be as good as this person, so there's no point even →

Left
800 LEDs are in this costume, all controlled by a single Arduino and one battery

Svetlana Quindt

INTERVIEW



Left
Svetlana runs KamuiCosplay.com with her husband Benni, who also does most of the paintwork (including this incredibly detailed costume from the *Monster Hunter* game)

trying". I don't know if the maker community is like that as well, but cosplayers easily get discouraged because they see all these crazy builds and they expect that their very first costume needs to be just as good as that. I try to keep reminding people that all of us, we all start small. I like to keep sharing my very first costume from time to time to show them that my first costumes were also bad, and that's totally OK, because you have to grow, you have to try new things. That's really important to remember.

I wrote two books about electronics as well: one for basic LEDs and one for advanced Arduino programming. They are a great start for people who have no idea, who want to try out something, and also point out that there are some cool things you can try out. It's actually not that hard – the hardest thing is getting started.

HS Speaking of which, the Monster Hunter costume you made... how did you get all the lights working in it?

SQ I'm not really that much into electronics. I mostly follow tutorials. My very first light-up costume was actually a Protoss Wizard [from the Diablo game series], which I made in 2012.

I work mostly with the Adafruit Trinket 5 V. It's basically like a small Arduino. The good thing about Adafruit products is that they have very detailed tutorials that you can follow, and they also give you a direct link to all the hardware and software you need.

I can't really program. I learned how to edit programs; mashing different codes together and creating something new from that.

HS Some would say that is the essence of programming: taking what's already there and making it work for you.

SQ That's important. I'm never trying to create a circuit, or write code: I'm always trying to create a better costume, and the code is incidental. My electronics are very simple, very basic, but there are →

costumes on top. I think it's really cool to work with electronics, but you have to find a project where you can put everything in. Do you know Dragon Con? A few years ago, there was a guy who made a white suit completely filled with NeoPixels, and he was on the dance floor – his light-up suit was animated to the music. It was incredibly cool.

HS We had something similar in the magazine a while back – Geek Mom Projects' light-up dress, triggered by sound to animate some LEDs. It's a simple idea, but it was executed so well that it just looked amazing.

SQ It's not simple if you have no idea where to get started. This is why I wrote my books. I wrote my first LED book for kids, and I often get messages from parents. I got one from the parent of a seven-year-old girl who was really proud because the kid had started with electronics. It is really important to take the fear from people.

For you, it might be simple, but others might be worried about picking up a soldering iron because they are scared that they will burn themselves. What if you get the resistors wrong? Will it catch fire? Will it blow up? People can be very intimidated by electronics.

When I wrote my first book, I pre-calculated a few basic circuits up to twelve LEDs, so people don't even have to do any calculations; they just have to follow the schematic, solder everything together, and it lights up. When I write my books, this is very important – taking the fear away from people and making it very, very basic. As soon as you get them to the point where they realise they can light up one LED, then it works.

When I got into electronics, all the tutorials I found were written by engineers. They started with formulas and calculating voltage drops and resistance, and I thought 'oh my god, I can't do it'. My first circuit was a button cell battery, a 3 V battery – slide an LED onto it and then stick it in place with some tape. Put it in your costume, and you have a very simple cosplay add-on. That's so simple – anyone can do that.

HS One thing about cosplay culture that we were bemused to learn is that there's a philosophical divide between foam and 3D printing. What's that all about?

SQ 3D printing is a very exciting topic. We waited a few years before tackling it because I thought that the technology was simply not ready yet. It was too expensive, and now it has reached a point where a good-quality printer costs just as much as a sewing machine these days. It's a really cool technique, but the mentality of a lot of people in the cosplay community is still a little bit dismissive of it. 'It's not handmade; you just press a button, and then your costume is done'.

> A good-quality printer costs just as much as a sewing machine these days

There's still a bit of bias against it, so we try to correct this a little bit, by showing people who are interested how to get into it, and it's actually not just pressing a button – it still involves a lot of work, but it also has a ton of potential.

People connected with cosplay are always stressing the importance of handcrafting things. It's stupid because if you took that to its conclusion, then you shouldn't use a sewing machine, you should stitch everything by hand. It will change in time, especially when people actually see how much work goes into 3D printing as well. Sure, you can download a ready-made model, print it out, and paint it no matter how rough the surface is. However, at the same time, when you do this in a costume competition, you simply don't get points for this part, because no craftsmanship was involved in making this part – you'll only get points for the paint job.

It's really exciting because now more and more people are getting into 3D printing. I think, also, one reason is that Blender 2.8 has apparently become much more user-friendly. My husband hated Blender before, but now he loves using it. We use Blender to create armour patterns. I have a 3D scan of my body imported into Blender. We can sculpt an armour design in software, transfer that onto paper to make a template, and then use that to trace the shape in foam.

So we really try to combine different technologies to create a costume. In the past, I think it was more delineated: making with foam, making with sewing, etc. Now cosplayers who compete in costume competitions try to combine as many techniques and materials as possible.

As a cosplay competition judge, I simply say that if you master several techniques, you get more points than if you master just one technique or material. It's really exciting to see how many technologies and materials they combine just to create one single costume. It's insane.

HS It's certainly impressive. To have 3D printing, foam, electronics all in one build is a massive amount to get your head around.

SQ I also know people who got into the movie industry through cosplay. They started as cosplayers and used their costumes as a portfolio to get into film special effects. They work for Weta [Digital] or any other big studio movie companies' special effects. Isn't that amazing? I mean, sure, you can go to a costume design school and learn a few skills, but if you're making costumes, every single project requires new skills, new materials – people improve →

Above
Svetlana in character as herself, with faithful hound Zelda

themselves over the years like crazy. They learn so much – sure they probably will never be as good as a professional sculptor from Hollywood who does only sculpting, but it's crazy that people can recreate the costume from Hollywood without the whole team, for example.

When the movie *Cinderella* came out, the studio must have had X number of people working on it for X number of months. And then a cosplayer hand-finished a replica of the dress in only a few weeks after the film was released. That's amazing.

> It's not about making a cool costume; it's really about failing and crying, but not giving up

HS People love their passions. As you said, when you go from being a consumer to a participant, that's a big mental shift.

SQ It's a big challenge to try something new. Often there are no tutorials, nobody to take you by the hand. None of these cosplayers are professionals; they didn't learn sewing, sculpting, and it's not like any of us went to costume-making schools. Everything I do, all the skills I do right now, I learned them all by myself. I learned everything simply by failing, by crying in the middle of the night, wanting to throw everything away in pure desperation. This is really what cosplay is about: it's not about making a cool costume; it's really about failing and crying, but not giving up, just keep on doing it.

Sometimes I'll be working on a costume for a whole year just because I'm too dumb to get it finished. There are no tutorials to follow, and I'll be sitting in front of a half-finished costume with the deadline coming closer and closer. You have a deadline, and you want to finish this damn thing.

I don't know if you saw my Nova photo shoot – the costume with the light-up ponytail? This costume includes 800 LEDs, which are run with a single Arduino with a single LiPo battery. I have no idea how much a LiPo battery can withstand, but to do the photo shoot, I was running 6 or 8 amps. I wouldn't recommend doing this. The question was running through my mind: what if something burns down?

I always try to include safety precautions, safety notes, and everything in my books, but I know I'm not a professional, so I always get my books checked by engineers; people who know what they are doing. I know someone who works for Hollywood Studios who does all the lights effects and everything, and they help a lot. I've learned so much just by trying things and seeing what happens.

Cosplay is such an amazing hobby that teaches you so many things in your life. Not to let other people say that you can't do something, or be scared about something that you've never done before. I learned English with cosplay, and after I started with cosplay, I got much better at job applications – I became much more self-confident. I was scared in school just talking in front of 20 people in my class, but now I have no problem talking in front of 20,000 people.

Cosplay changes so many things in your life, improves so many things... I don't know. I love it. It changed my life. I wouldn't be where I am today. It was the best decision of my life, and I'm really happy that I have my husband and I can do this full time. □



Right
Building up layers of detail with the Worbla modelling material

FEATURE

IMPROVISER'S TOOLBOX

PAPER

The seemingly simple material can be quite a page turner



Mayank Sharma

@geekybodhi

Mayank is a Padawan maker with an irrational fear of drills. He likes to replicate electronic builds, and gets a kick out of hacking everyday objects creatively.

For aeons, paper has been the storage medium of choice for transfer of knowledge from one generation to another. We began documenting stuff on a paper-like material that was produced from the extract of the papyrus plant. The Egyptians mastered the art of creating papyrus paper around 2500 BC, though the oldest surviving record of this material dates to around 2200 BC.

In 100 BC, the Chinese started creating paper from hemp during the Western Han dynasty. Hemp continued to be the raw material of choice for making paper for several centuries, though the source of the hemp changed over the years. It wasn't until 500 AD that hemp made way for other materials, when the Mayans began using paper made from tree bark.

THE PROCESS OF MAKING PAPER WAS AN INVOLVED ONE, AND THE MACHINES COULD ONLY PRODUCE PAPER SHEET BY SHEET

Paper made its way to Europe in the 13th century and, by the 17th century, all the major European countries had set up paper mills. However, the process of making paper was an involved one, and the machines could only produce paper sheet by sheet. It all changed in 1799, when French engineer Louis-Nicolas Robert patented a continuous papermaking machine. Thanks to a variety of reasons which would only be of interest to history buffs, the first such machine was built in Hertfordshire, England, in 1803.

There are several types of paper available these days, with wood-pulp paper being the most commonly used kind in the world. But there's a conscious effort to move to more ecologically friendly paper. For instance, there's wood-free paper that uses chemical pulp instead of wood-pulp. Then there's what is known as tree-free paper, which has the least amount of negative impact on the environment.

Paper can also be classified by its use, such as printing paper, writing paper, drawing paper, blotting paper, and more. Then there's paper that's created to serve a specific purpose, such as tissue, cigarette paper, and others. The most common classification though is by size. 'A series' paper ranges from A0 to A8, with A4 (210 mm by 297 mm) being the most common. There's also a B series that is mostly used for things like movie posters, books, passports, and such. Then there's the C series, which is used only for envelopes. As with virtually all measurements, North America doesn't use the ISO 216 standard paper sizes like the rest of the world. Instead, the US, Canada, and Mexico use the Letter, Legal, Executive, and Ledger paper sizes.

No matter what paper size you use, here's how you can fold, tear, and tape it creatively.

GIF PLAYER

A self-confessed hoarder of items she wants to upcycle, Janet's happy place is her crafting corner. She uploaded the Instructable for her GIF player just before Valentine's day, and suggests it'd make for a unique gift. This is especially true when you consider the fact that the build really doesn't require any more materials than a greetings card, namely paper and glue. A major part of the instructions involves fabricating the eleven different components from sheets of paper, and then putting them together to create the device. To help with the construction, you can download the templates from her Instructable and use them as guides. Janet believes that the best Instructables are the ones that make replicating them fairly straightforward. For some bits, she's used a laser cutter. If you don't have one handy though, you can use Janet's templates and then cut them out with scissors and skewers. The next major task involves creating the individual frames for the animation. Here again, Janet shares her simple recipe for transforming videos and live photos into a set of images using an iPhone. At the end, simply download the frames as individual images and print them, before gluing them to the individual frames of the device following her very detailed instructions. →

Project Maker
JANET FOX

Project Link
hsmag.cc/1bSjUN



Left ←
Janet's GIF player is actually a form of a Mutoscope, which was one of the picture devices to give the illusion of motion

FEATURE

PASSIVE PHONE SPEAKER



Speaker docks for your smartphone are pretty common. But Rob's version, made using paper and shaped like the horn of a phonograph, is sure to turn heads. Rob's shared two designs for the fully functional dock depending on the placement of the speakers on your phone. The iPhone design works for phones that have the speakers at the bottom, while the Galaxy design works for devices that have the speakers at the back. Begin by downloading Rob's design that he's shared in his Instructable, and print it on coloured photocopier card. Make sure the paper isn't too thick. For reference, Rob's used 230 micron-thick card-stock. The horn is the same for both designs, and Rob suggests printing the horn parts of the dock on paper in two complementary colours. Once you've printed out the parts, score along the dotted and dashed lines as instructed in the PDF, before carefully cutting them out. Then follow Rob's Instructable to assemble the pieces and glue them together as illustrated. You might have to adapt the design depending on the exact position of the speakers on your phone, but this isn't much of a chore since Rob has designed the base to be usable with a wide range of phones.

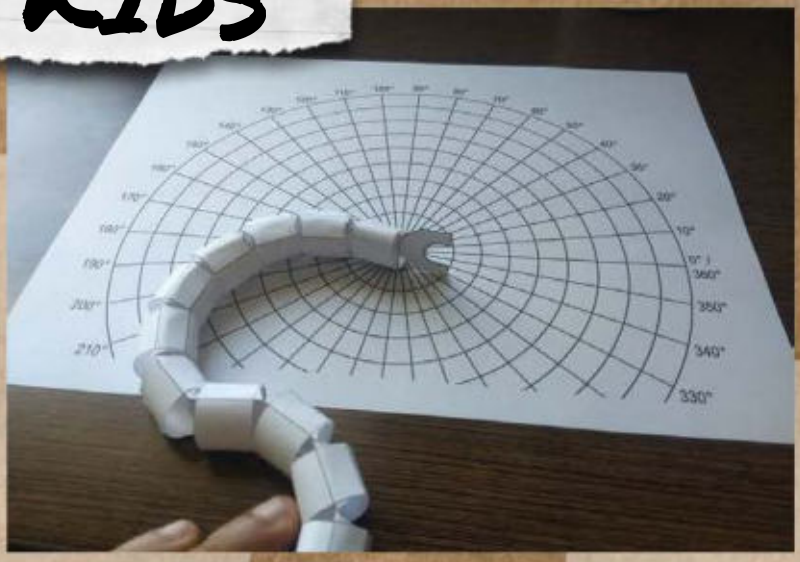
Project Maker
ROB IVES

Project Link
hsmag.cc/ECVVJQ

YOU MIGHT HAVE TO ADAPT THE DESIGN DEPENDING ON THE EXACT POSITION OF THE SPEAKERS ON YOUR PHONE

Above
If you're looking to get creative with paper, make sure you check out Rob's portfolio of paper toys

ANIMATRONIC ARM FOR KIDS



Rudrarup Mitra wanted to show his students how the different parts of a drive-by-wire animatronics machine work. The prolific maker is a techy by training, who now runs STEM workshops along with his wife in Pune, India. To create the animatronic arm, grab the PDF templates he's shared, and print them out. Then follow his Instructable to cut, fold, and glue the various components of the arm. Once it's completed, you can tug on the controller strips to flex the animatronic limb. The gripper at the end is ornamental, although at one of his workshops students wondered if they could add more control strips so as to use the grippers to grab on to something: "One particular kid mentioned we can break open a relay and add the solenoid. That made me think: kids nowadays still have innovative skills that we fear are getting lost due to the newer study patterns."

Project Maker
RUDRARUP MITRA

Project Link
hsmag.cc/jlzwWC

Above
You can use the circular calibration graph to practise and master the control of the arm

FLOWER-POTS



To do her bit to reverse environmental depletion, Regan, a Girl Scout, used old newspapers to create flower-pots. Her technique is fairly straightforward: Tear the newspapers in half and fold each in half lengthwise, like a hot dog. Then open them up and fold the sides into the crease. Repeat the process a few more times until you have folded the paper into a tight strip. You'll need multiple strips to create a pot, depending on the size of the pot. Regan suggests using around 30 strips for a pot the size of a coffee cup, while larger ones such as one to use for a houseplant will need up to 200 strips. Once you have the strips, follow Regan's instructions and wrap them tightly into a coil with some tape. Then gently push them from the middle to create the pot. Regan cautions you to not push too hard, else the coil will unravel, and you'll have to wrap it all over again. ◘

Project Maker
REGAN SCHMIDT

Project Link
hsmag.cc/2O4wKD

Below
The pots are coated with Mod Podge to make them more sturdy, but you can also use PVA glue instead





Above
Two more masks being made

Protecting the NHS

A group of makers creating face masks for health workers

By **Ben Everard**

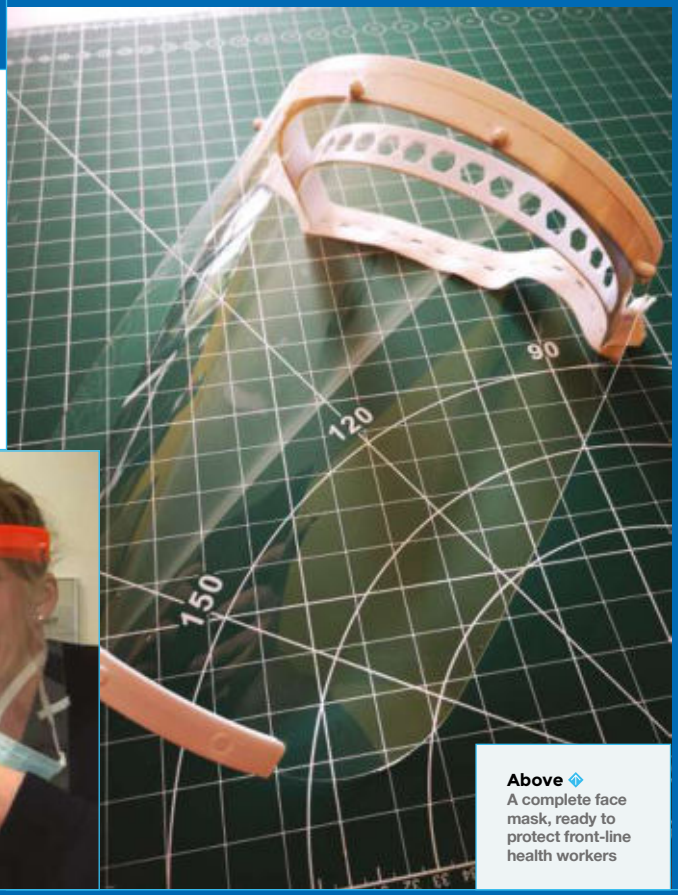
The first half of 2019 will be forever remembered as the time COVID-19 rampaged around the world. It's been a time of suffering and loss for many. However, hopefully, it'll also be remembered as a time people came together to help the most vulnerable, and a time when community spirit helped limit the damage that the disease caused.

Many groups have rallied together – we don't have time to thank them all here, but we would like to take a look at how a group of makers in the UK have been helping the National Health Service (NHS) in a time of equipment shortage.



Credit 3DCrowdUK volunteers

Right
Front-line staff are short of PPE, so every bit is vital



Above
A complete face mask, ready to protect front-line health workers



Left
Maybe some bright colours will cheer up someone's day

SHIELDS UP

Quite early on in the disease's spread, it became apparent that a lack of personal protective equipment (PPE) would be a significant problem for health workers around the world. Some PPE requires special materials and processes to make, but face shields can be made with common maker equipment – laser cutters and 3D printers. There are several designs around that are being used in healthcare around the UK, and each is suited to a different set of available tools. A few of the options are:

- Smoke & Mirrors, along with Kitronik, have devised a shield that can be quickly produced using just a laser cutter: **smokeandmirrors.store/blogs/news/face-shields-for-covid-19**
- 3DVerkstan have designed a shield that you can 3D-print, where the transparent guard is made from an A4 OHP film cut with a hole punch: **3dverkstan.se/protective-visor**
- Prusa Printers modified an existing design to print quickly and be more comfortable to wear: **prusa3d.com/covid19**

VENTILATORS

The world also has a shortage of ventilators – machines that help the most critically ill people to breathe. While there has been a huge maker effort to try to design models that can be easily built, at the time of writing there aren't any designs that have reached a high enough standard to be suitable for use. However, this may change in the coming weeks and months.

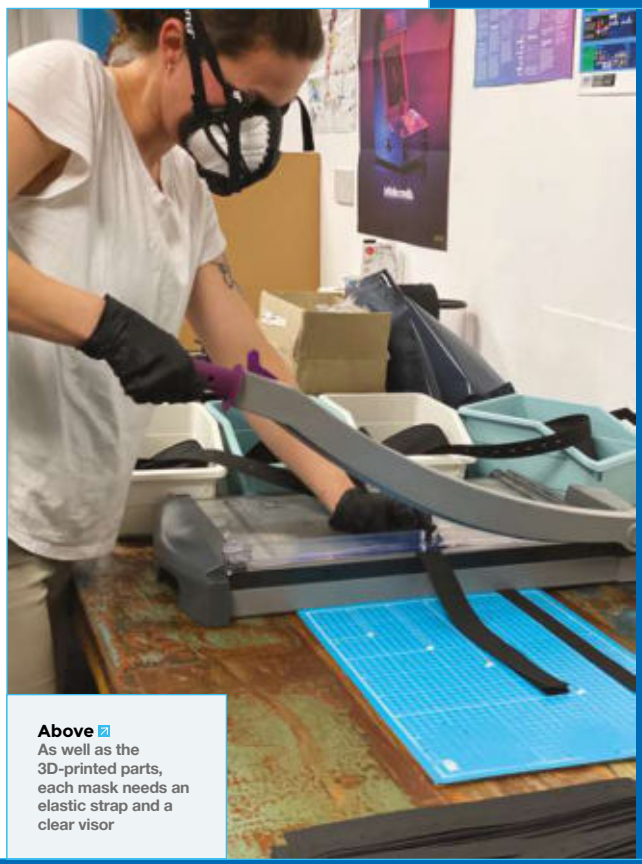
If you have a 3D printer and can help, sign up at 3dcrowd.uk

Here in the UK, the largest group effort is co-ordinated by 3DCrowd UK, printing and distributing the Prusa design of shield. Over 5500 volunteers (including the staff of HackSpace magazine) are now working on producing these masks for distribution to front-line medical staff.

They also have a crowdfunding campaign to help purchase filament and visors to ensure people can keep making as many shields as possible. DPD helped gather the parts from people printing them, and Pimoroni acted as a central hub to gather, assemble, and distribute them to hospitals around the UK.

If you have a 3D printer and can help, sign up at **3dcrowd.uk**. If you don't have a printer, but would like to contribute, you can support the project at **gofundme.com/f/3dcrowd-emergency-3d-printed-face-shields**.

We still don't know how this pandemic will end, but we do know that makers have already helped thousands of medical workers around the world, and will continue to do so for as long as the pandemic runs.



Above
As well as the 3D-printed parts, each mask needs an elastic strap and a clear visor



AROUND THE WORLD

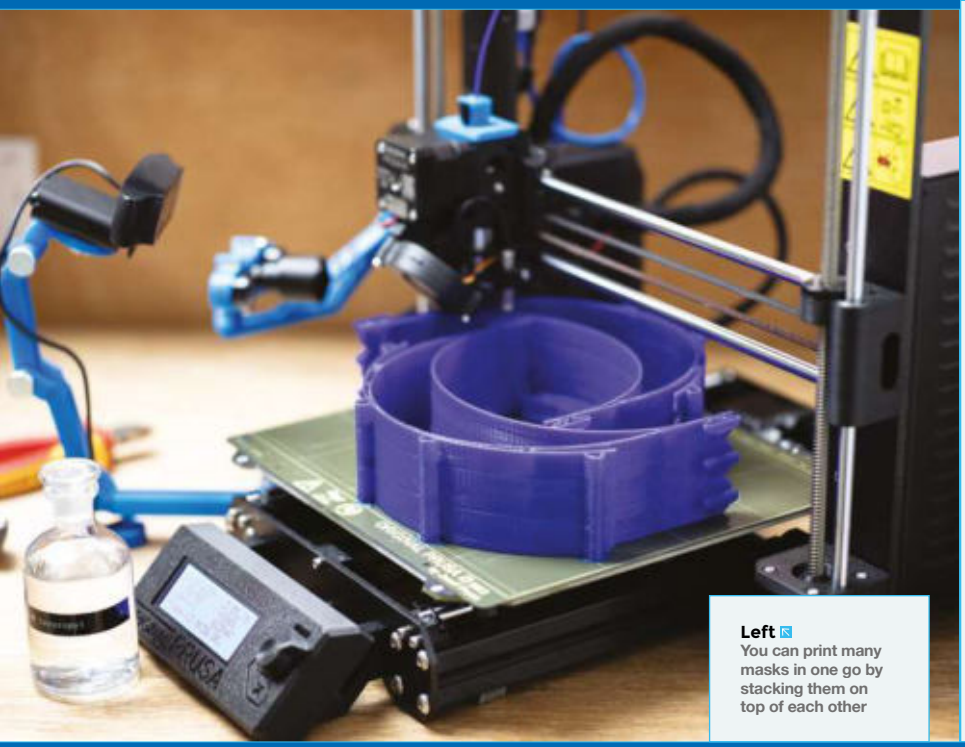
While we're looking at an effort here in the UK, similar maker groups have formed around the world to focus on the particular needs of their health providers. If you're in a position to help, keep an eye out in the usual maker channels to see what's going on in your area. This Prusa printer's blog post is a good starting point: **prusa3d.com/covid19**.



Right
Mask parts are bagged immediately after being printed to help keep them infection-free



Above
Eight more masks to protect those fighting the disease



Left
You can print many masks in one go by stacking them on top of each other

PITCHING IN

Maker companies around the globe are helping join the fight – some of them in ways they can't yet talk about publicly for a host of reasons. It's far too early to give any form of exhaustive rundown of who's doing what, but we can at least celebrate those who we know are doing excellent work.

As well as the organisations mentioned in the main text, Adafruit deserve a big thanks. They're based in New York, which has been particularly badly hit. However, through this, they've turned from manufacturing hobbyist electronics to making PPE as well as parts for medical equipment and other bits and pieces that are needed to keep hospitals running smoothly.

Hackster.io have partnered with the UN Development programme to help makers detect COVID-19, protect communities from it, and reduce the disease's economic impact. Take a look at covid19detectprotect.org/ for full details.

As the situation is changing rapidly at the moment, there will be many others that we've missed out (or whose impact only becomes clear in the coming weeks and months). As things become clearer, we'd like to continue to highlight great work that's being done. If there's someone you know doing great work, get in touch with us – hackspace@raspberrypi.org – as we'd love to feature them in the future.

Wireframe

Join us as we lift the lid on video games



Visit **wfmag.cc** to learn more

HackSpace
TECHNOLOGY IN YOUR HANDS

FORGE

HACK | MAKE | BUILD | CREATE

Improve your skills, learn something new, or just have fun tinkering – we hope you enjoy these hand-picked projects

PG 86
FIRE IN A JAR
All the ambiance of a candle, but without the fire risk

PG 90

3D-PRINT WOOD
Get a more natural look for your automatic creations

PG 94

MAGNETS EVERYWHERE
Find out how these iron attractors can be put to use

PG 98
LOW-POWER RASPBERRY PI
Keep your sensors running for days on a battery

PG 80
SCHOOL OF MAKING
Start your journey to craftsmanship with these essential skills

80 CircuitPython ulab
82 micro:bit radio



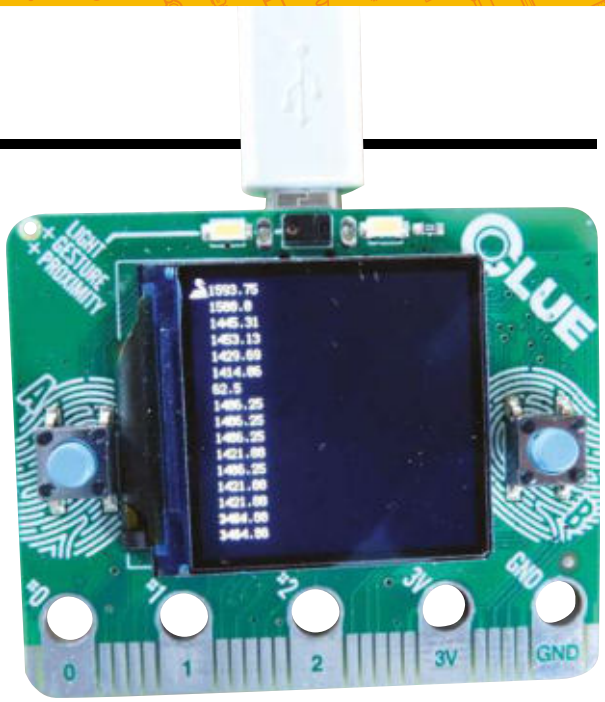
PG 102
DIY VACUUM FORMER
Quickly suck molten plastic into (almost) any shape

PG 108
AFFIRMATION MIRROR
Inject some positivity into your morning routine

PG 112
3D-PRINTING SUPPORTS
Investigating Cura's experimental supports

CircuitPython ulab

Fast mathematics for tiny microcontrollers





Ben Everard

@ben_everard

Ben loves cutting stuff, any stuff. There's no longer a shelf to store these tools on (it's now two shelves), and the door's in danger.

Above
A frequency meter can be used to tune musical instruments, or even tell the speed of a power tool

Python is fine for many uses, but when it comes to number crunching, it struggles. For regular Python, designed to run on full computers, the solution comes in the form of NumPy and SciPy. These libraries contain number-crunching code that runs much quicker than is usually possible. This is because Python can work with code written in other, much faster, languages.

While NumPy and SciPy don't work in MicroPython and CircuitPython, there is a similar library that does – ulab. We'll look at the CircuitPython version here.

It's only available on the very latest release – 5.1 rc0 – or higher, so you'll need to upgrade to this. Also, it's only compatible with Cortex-M4 ARM chips at the moment, which mostly means those boards based on SAMD51 or nRF52 series microcontrollers. You can download the latest firmware for any CircuitPython compatible device from **circuitpython.org**. We've used the Adafruit CLUE for this code as it has both a built-in microphone and screen, but you can use the same technique with any data that varies over time.

DECONSTRUCTIVE LEARNING

Let's explore the library by pulling an example apart and putting it back together differently. We'll start with the waterfall spectrum analyser: **hsmag.cc/Jnf4OC**. This produces a good-looking graphic based on audio data, but what is it doing and how is it doing it?

The guts of the program can be summed up in three lines, although they need a few bits around them to make them work:

```
mic.record(samples_bit, len(samples_bit))
spectrogram1 = ulab.extras.spectrogram(samples)
graph.show(data)
```

The first line grabs some data from the mic; the second performs a ulab spectrogram operation on this data, and the final displays this spectrogram.

Spectrograms are special forms of fast Fourier transforms (FFTs) – these are a way of extracting the frequency data from time-series data. The process with which it does this is fiendishly mathematical and includes the concept of complex numbers (which contain both 'real' and 'imaginary' parts). We'll ignore all this as, for our purposes, the output is more important than the calculation.

The output of a spectrogram is another array, but one that describes the frequency data of the sample. If you consider that all noises contain signals at lots of different frequencies at lots of different amplitudes, the spectrogram deconstructs this. The first data point in the array contains the amplitude of the signal at a frequency that's one waveform in the sample. The second data point represents two waveforms in the sample, and so on. In order to make sense of the data then, we need to know what these frequencies are.

> "The process with which it does this is fiendishly **mathematical and includes the concept of complex numbers**"

The mic takes samples which contain mic readings at a rate of 16,000 times per second, and in the default code, it takes 256 of them in one sample. Each sample is therefore 256/16,000 seconds long (0.016). The lowest frequency that can be completely captured in the sample has a wave that's exactly 0.016 seconds long, and that's 62.5 Hz.

It can be useful to know what is the main frequency in an audio sample, so let's take a look at how to display this.

Add the following immediately below the line `max_curr = ulab.numerical.max(spectrogram1)[0]`:

```
        index=0

        for item in spectrogram1:
            if item == max_curr:
                print((index+1)*(16000/fft_size))
            index=index+1
```

This simply finds the location of the largest value in the sample and translates that into a frequency.

The slight problem with this is that it'll only give you a frequency to ±31.25 Hz (as readings are in multiples of 62.5 Hz), which isn't very accurate. The solution is to increase the `fft_size`. We found 2048 proved to be a nice balance between accuracy and speed (longer samples take longer to both record and process).

You can also remove all of the graphics code and just display the number on the screen. The code for this is at **hsmag.cc/OjSHxY**.

RETRO DISPLAY

Let's look at the graphics display. In the original code, the `RollingGraph` class has a show method that takes the current data and displays it on a new line, mapping each value to a colour so higher values are mapped to lower colour frequencies (i.e. closer to red in the spectrum). This gives a good overview of how frequencies in the sound are changing over time, but let's look at a more traditional audio spectrum display – a bar graph. In our bar graph, we'll include the current data and the previous data so we can see how the frequency has changed between two samples.

The full code is at **hsmag.cc/6K6knl**, but the key code is the new `BarGraph` class, which is:

```
class BarGraph(displayio.TileGrid):
    def __init__(self, scale=2):
        # Create a bitmap with heatmap colors
        self.bitmap = displayio.Bitmap(display.
width//scale,
                                       display.
height//scale, len(palette))
        super().__init__(self.bitmap, pixel_
shader=palette)

        self.scroll_posn = -1
        self.scroll_display = [[],[],[]]
        self.colours = [3,11]

    def show(self, data):
        bitmap = self.bitmap
        if self.scroll_posn == -1:
            self.scroll_display[0] = data
            self.scroll_display[1] = data

        else:self.scroll_display[self.scroll_posn]
= data

        board.DISPLAY.auto_refresh = False

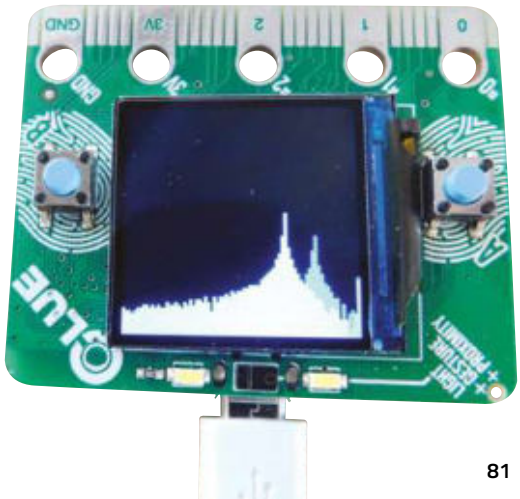
        for colour in range(2):
            for x in range(min(bitmap.width,
len(self.scroll_display[0]))):
                for y in range(bitmap.height):
                    if self.scroll_display[((self.
scroll_posn+colour+1)%2)][x] < y:
                        if colour == 0:
bitmap[x,y] = 100
                    else: bitmap[x,y] = self.
colours[colour]

        board.DISPLAY.auto_refresh = True

        self.scroll_posn = (self.scroll_posn + 1)
% 2
```

This creates a bitmap that you can update with `bitmap[x,y] = 100`. The colour value corresponds to that in the palette – we've used the palette from the original waterfall sketch. This works in a really simple way: it stores data in an array (`scroll_display`), and `scroll_posn` keeps track of which is the current 'live' data and which is the previous data. We then iterate through every point on the screen twice. First, for the historical data, we draw the full graph and background (hence the `if colour==0` statement); the second time, we just draw the graph bars (not the background). This overwrites some parts of the previous graph, but any peeking out beyond the second graph will still be displayed because we didn't overwrite the background.

Hopefully, you now have a better idea of how ulab works, and how to display data on an Adafruit CLUE. Also, we've shown how to poke about with example code to learn more about how a library works. Now it's your turn to pull apart the code and make it do what you want. ◻



Below
This author's whistling now brought to you in image form

Working with micro:bit radios

The built-in radios on micro:bits are great for lots of remote control and communication projects – let's take a look!

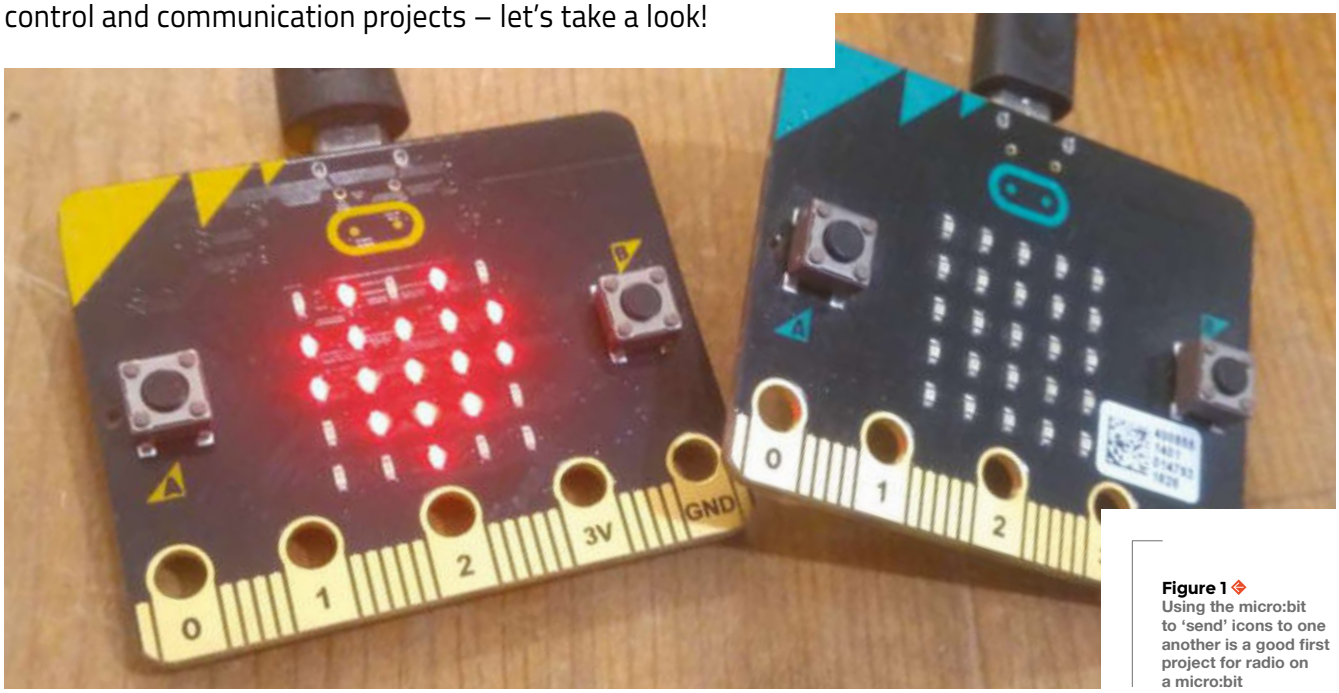


Figure 1 Using the micro:bit to 'send' icons to one another is a good first project for radio on a micro:bit



Jo Hinchliffe

@concreted0g

Jo Hinchliffe is a constant tinkerer and is passionate about all things DIY space. He loves designing and scratch-building both model and high-power rockets, and releases the designs and components as open-source. He also has a shed full of lathes and milling machines and CNC kit!

Since it arrived in around 2016, the micro:bit has established itself as a very capable and fun platform to play with and learn. In issue 29, we reviewed the Kitronik :MOVE mini MK2 robot kit which uses a micro:bit as its main control board. It's inspired us to explore the radio features of the micro:bit, and to do some projects using remote control.

The radio on micro:bit can be set up as Bluetooth LE. As such, one of their primary uses is to flash code to a device. This is handy if you are using a phone or a tablet to program the micro:bit, or indeed if your hardware project has escalated to a point where it's hard to access the micro:bit USB port. However, in this tutorial, we are focusing on some simple activities to show how we can use the radio aspects to enable multiple micro:bits to communicate and interact.

TUNING IN

As a simple first radio project, we are going to get two micro:bits to communicate and send each other commands to show, or not show, an icon (**Figure 1**). We are going to create one program and flash it to two micro:bits. Open the online MakeCode editor and start a new project. As a default, a new project should have an 'on start' and a 'forever' block on the stage. Navigate to the Radio menu, select the 'radio set group' and set it to a number. As both micro:bits will be flashed with the same project file, this ensures that both micro:bits will be on the same 'channel'. As seen in **Figure 2**, we set the 'radio set group' to '1'.

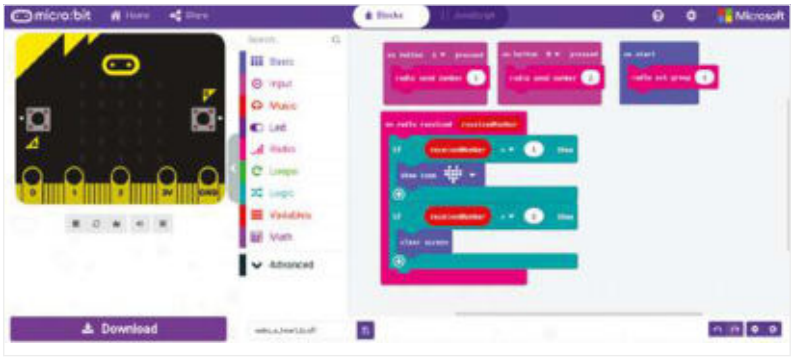
Next, move to the Input menu and drag two instances of the 'on button pressed' block onto the stage. Set one of these blocks to button 'A' being pressed, and set the other to button 'B' being pressed. Then, inside each block add a 'radio send

MAKE CODE



For the examples in this article, we have used the online block-based MakeCode editor, which is available online at **makecode.microbit.org**. Dragging blocks and connecting them on the block stage to create the small programs is straightforward – we can save the code using the dialogue box at the bottom of the screen, naming them, and automatically downloading them. Connecting a micro:bit via a USB cable makes the micro:bit appear as a connected drive device. Then, copying and pasting the downloaded file onto the device flashes the code to the micro:bit. For a few seconds, the yellow activity LED will flash – once completed the micro:bit will disconnect as a USB drive before reconnecting, but the HEX file we placed in the 'drive' will have disappeared. On the MakeCode editor, if you close your browser, but return to the site later, you should have your previously saved projects cached and available to reopen. However, you can also re-upload the HEX files you downloaded to the editor – they will reopen as a project.

number' from the Radio blocks menu. Set this block to broadcast the number '1' over the radio for when the A button is pressed and the number '2' for when the B button is pressed. This means we have created a simple set of transmit instructions for the micro:bit to broadcast those numbers on that radio group. For the receiver side of the project, we are going to set up the micro:bit to listen out for those



numbers, and describe the actions to take when it receives those numbers.

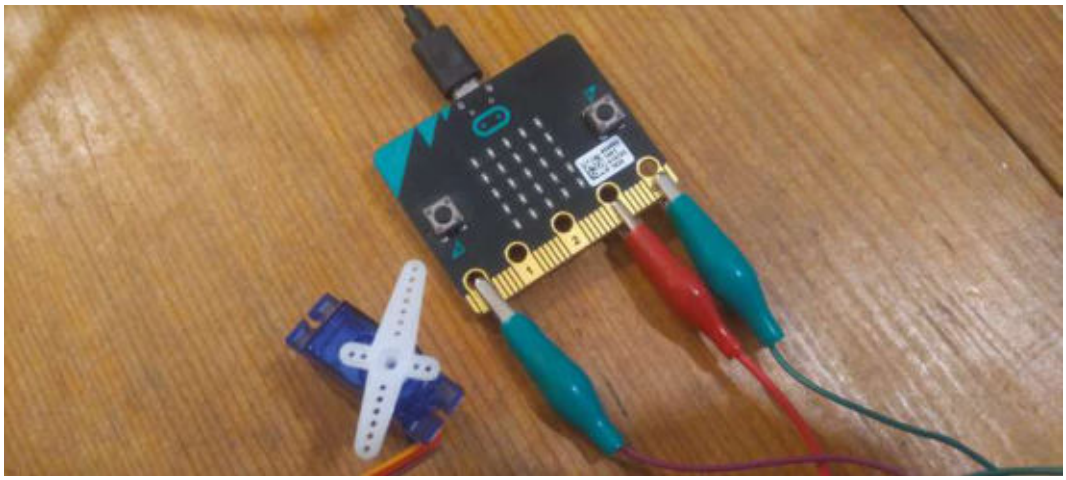
From the Radio menu, select an 'on radio received' block and drag it into the project. It should, by default, have 'receivedNumber' in the pill-shaped area of the block. Next, from the Logic menu, drag an 'if-then' statement block, which should be the first block listed

> We are going to get two micro:bits to **communicate and send each other commands**

under 'conditionals'. Place it inside the 'on radio received' block. In the Logic menu, again, select the first diamond-shaped block under the Comparison sub-menu which, when you hover over it, should be described as 'return true if both inputs equal each other'. Drag this comparison block and drop it over the current diamond-shaped area in the 'if-then' →

Figure 2
The blocks needed to make our first radio project – controlling micro:bit to turn a heart icon on and off

Below
The micro:bit is capable of driving a micro servo directly. Here, we have connected the ground and power pins, and the yellow servo signal wire is clipped to P0 on the micro:bit



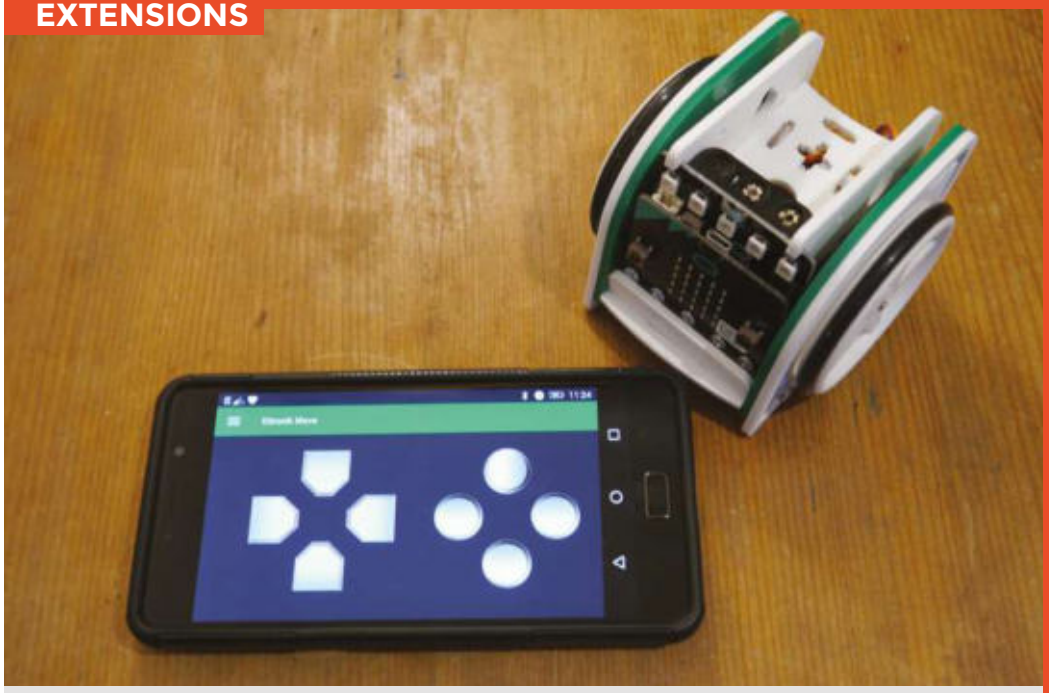
YOU'LL NEED

- **Laptop and internet connection**
- **micro:bits** (optional)
- **Kitronik :MOVE mini MK2 robot kit** (optional)

QUICK TIP

No hardware? Don't worry. If you don't have a micro:bit, you can play with these ideas in the MakeCode editor, and it emulates the project hardware on screen – it even will add servos and show them rotating!

EXTENSIONS



Kitronik have a set of extension blocks that can be added to the MakeCode environment for the :MOVE mini MK2 robot which can be found by browsing the Extensions page in the Blocks menu. This block collection has numerous blocks designed to work specifically with the :MOVE mini hardware. As the micro:bit radio can be configured as Bluetooth LE, they (and others) have also published phone applications that can act as a controller, meaning you only need one micro:bit, which you can interact with or control via your phone. There are numerous Bluetooth controller applications out there – this can be a fun and useful way to create projects.

statement block. Next, click and drag the 'receivedNumber' pill-shaped block in the 'on radio received'; this should create a copy of that variable, and drop it into the left-hand side pill shape of the comparison block you just placed. On the other side of the comparison diamond, select the number '1'. This 'if' statement is now set to do something when the received number matches the number '1' when the A button is pressed on another micro:bit. To finish off this 'if-then' statement, grab a 'show icon' block and insert it and set it to show the heart icon. To complete this project, repeat the above steps to set up another 'if-then' statement for when button B is pressed, setting this to clear the screen as seen in **Figure 2** on previous page.

Save and download this project; then flash it to two micro:bits – you should now be able to turn the heart icon on and off on each micro:bit.

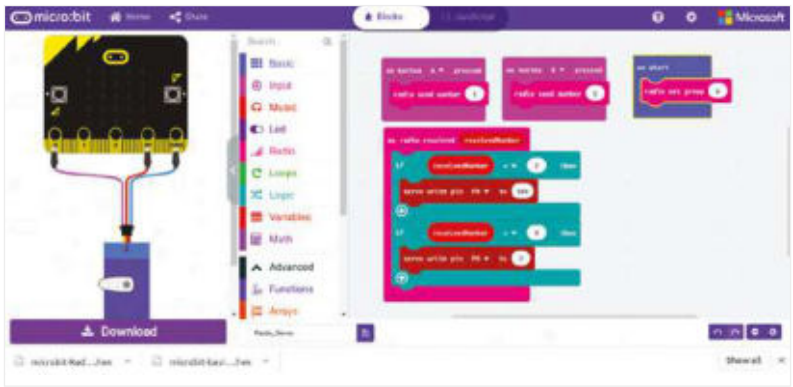


Figure 3 The MakeCode editor blocks for our experiment to control a servo from a separate micro:bit

MAKING THINGS MOVE!

For our next project, we are going to connect a micro servo directly to one micro:bit and control the servo position with another micro:bit. We are again going to write the 'transmit' and 'receive' parts of the system in the same project, so we will load the same file to both micro:bits. Much of this project is the same as the first project, so you can recycle that project and just make changes where needed. Set up the project

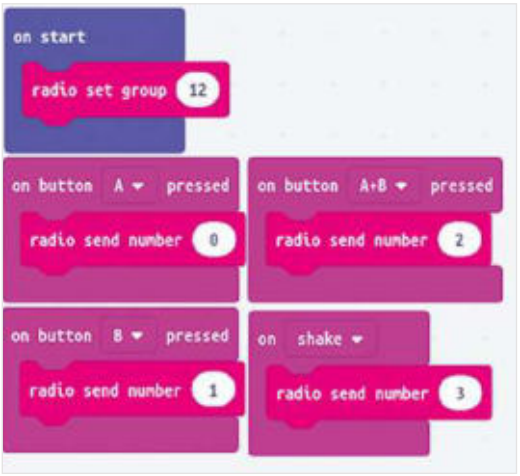
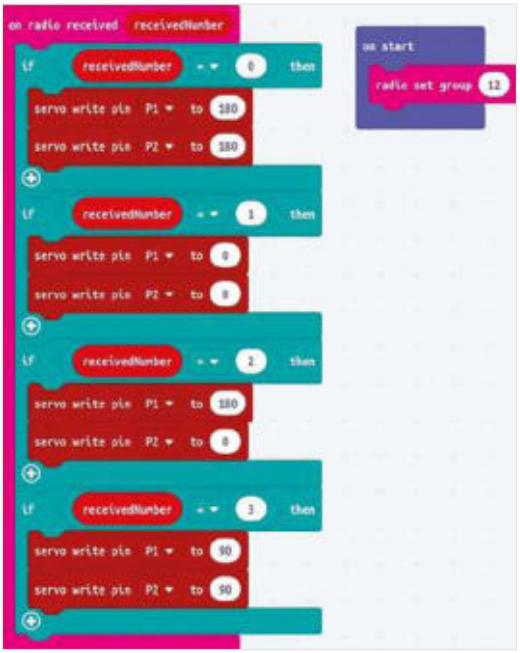


Figure 5
The blocks connected to form the program for the micro:bit acting as a transmitter/controller



in the same way – set the radio group number, and create blocks to broadcast different numbers when buttons A and B are pressed. Similarly, set up 'if-then' statements, but for the 'then' part of each statement, select a 'servo write pin' block from the Pins menu which is nested under the Advanced tab. As you can see in **Figure 3**, we connected the servo on pin 0 and set it to rotate 180 degrees when button A was pressed, and back to the 0 position when button B was pressed. Flash this to two micro:bits, one with a servo attached to pin 0, and you should be able to control the servo from the other micro:bit.

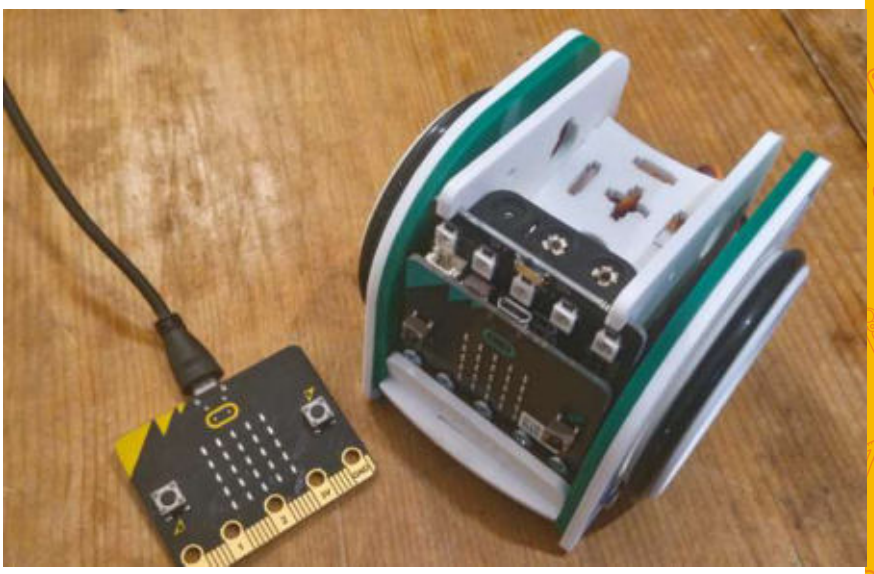


Figure 4
The Kitronik :MOVE mini MK2 robot set up with another micro:bit acting as remote control

As a final project for this tutorial, we wanted to use one micro:bit as a remote control for the Kitronik :MOVE mini MK2 robot (**Figure 4**). For this, we wrote two projects: one for the transmitting controller (**Figure 5**), and one for the receiving micro:bit in the :MOVE mini MK2 robot (**Figure 6**). We used the same format and approaches as the other projects in this tutorial, but added more instructions. On the controller side, we added four actions: buttons A and B being pressed separately, button A and B being pressed together, and finally, an 'on shake' block. We set each of those up to broadcast a different number. For the receive part, we set up similar 'if-then' statements for each radio 'receivedNumber' and added servo instructions to create forwards, rotate left, rotate right, and stop movement associated with each command. The :MOVE mini MK2 robot uses two continuous rotation servos, and as such, needs two 'servo write' blocks for each desired movement, which we had explored as part of the original build of the :MOVE mini MK2 robot. So, for example, both servos set to 180 makes it rotate to the left, 90 and 90 stops both servos (**Figure 6**). Having flashed both our micro:bits and connected our controller micro:bit to a small USB battery pack, it worked perfectly to control the micro:bit! ◻

> "We wanted to use one micro:bit as a **remote control for the Kitronik :MOVE mini MK2 robot**"

Figure 6
The blocks connected that form the receiving program flashed to the micro:bit in the Kitronik :MOVE mini MK2 robot

QUICK TIP

The micro:bit can supply just enough power to run a micro servo directly – great for an experiment, but you probably need to use a motor or servo driver board if you want to build a better robot!

TUTORIAL

Build a realistic fire-jar animation

Create the illusion of flames to add ambience without the risk of an inferno



Dan Cogliano

@cogliano

Dan is an engineer by degree, and he likes projects with e-paper, LEDs, and anything that looks magical.

There's a lot of magic that can happen in a Mason jar. Just think of children or your own childhood memories of collecting fireflies and other insects and putting them in a jar. The display of fire in a jar can be just as magical for both adults and children. This tutorial shows how to use an Adafruit Feather running an Arduino sketch and an Adafruit 8×8 NeoPixel NeoMatrix to simulate fire. Put all of this in a jar and you'll have something magical to cherish.

THE HARDWARE

The illusion of fire is created using an Adafruit Feather, an Adafruit 8×8 NeoPixel NeoMatrix (see **Figure 1**), and a wide-mouth Mason jar. The inner diameter of a wide-mouth Mason jar is slightly larger than the Adafruit NeoPixel NeoMatrix, allowing the board to just fit inside the jar (Note: the NeoMatrix is too large to fit in a regular-size Mason jar). The NeoPixel NeoMatrix fills a 16 oz/500 ml size wide-mouth Elite Mason jar nicely without a lot of space, although larger sizes can be used (see **Figure 2**



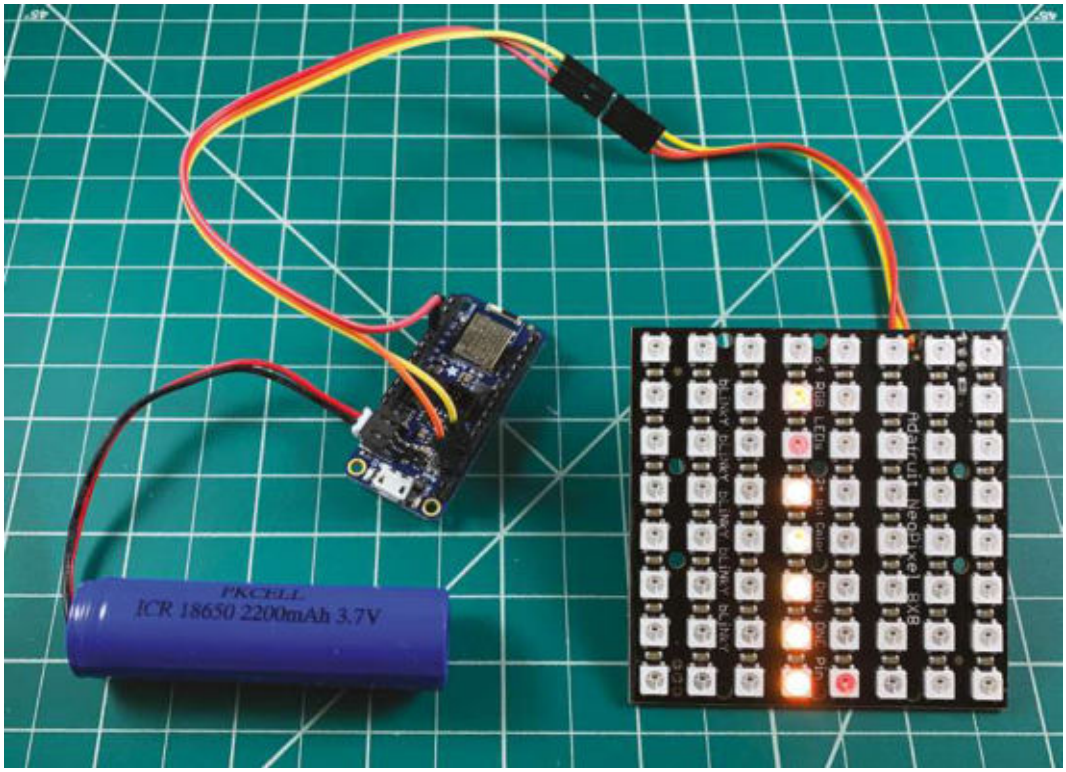


Figure 1
An Adafruit Feather and NeoPixel NeoMatrix are used to create the fire illusion (battery optional)

overleaf). Adding a LiPo battery allows its use without the need of a USB cable so the lid can be placed on the jar, making a self-contained fire in a jar.

Three wires connect the Feather to the NeoMatrix: power, ground, and DIN. The power pin connects to the Feather 3V pin, the ground pins connect to each other, and the DIN pin connects to the Feather's digital 6 pin (see **Figure 3** overleaf).

To make it easy to connect the NeoMatrix to the Feather, male or female headers can be soldered onto the back of the NeoMatrix. Be careful when soldering, as the soldering points are close to components on the top of the board (see **Figure 4** overleaf). Jumper wires can then be used to connect the bottom of the NeoMatrix to the Feather (see **Figure 5** overleaf). The type of jumper wires needed depends on which headers are used for the NeoMatrix and the Feather. This tutorial shows male headers soldered onto the NeoMatrix and female headers on the Feather. This method allows using either male/female jumper wires, or combining female/female jumper wires with male/male jumper wires.

THE CODE'S ON FIRE

The ZContent_ParticleFX library is an Arduino library created specifically for low-resolution boards like the Adafruit 8×8 NeoPixel NeoMatrix, although it can be extended for other pixel dimensions or higher-resolution displays. This library is used to animate points of light, or particles, in a matrix. In addition →

ADDING **THE MAGIC**

Diffusion is key to making the animation look realistic. Some diffusion ideas include using polyester fibre filling (used for pillows), a thin layer of Styrofoam, or painting the jar with semi-transparent spray-paint.

Semi-transparent spray-paint makes an excellent diffuser. However, it can be scratched off quite easily when applied to the outside of a glass jar. As an alternative, spray the interior walls of the jar so that the diffuser paint will not get scratched off during regular handling. Use masking tape on the outside of the jar to keep the paint off when spraying the inside.



YOU'LL NEED

- **Adafruit Feather with M0 or nRF52840 processor** (Use the nRF52840 to add Bluetooth features – adafruit.com/product/4062)
- **Adafruit NeoPixel NeoMatrix 8×8 – 64 RGB LED Pixel Matrix** (adafruit.com/product/148)
- **Wide-mouth Mason jar** (16 oz/500 ml or larger)
- **Jumper wires**
- **LiPo battery compatible with the Adafruit Feather** (optional)



Figure 2
The 71×71 mm Adafruit NeoPixel NeoMatrix fits nicely in a 16 oz/500 ml size wide-mouth Elite Mason jar, which measures about 95 mm/3 ¾" high

to fire, several animation ideas came to mind when creating this library, including fireflies, fireworks, falling snow, and a Tinker Bell fairy animation. All of these animations are possible with this library.

SMOOTHING THE ANIMATION

In order to make the animations look realistic in a low-resolution display, the library uses a technique similar to displaying images and fonts on a screen, called anti-aliasing. Anti-aliasing smooths jagged edges on a screen. With this library, drawing is done on a virtual 64×64 bitmap and then reduced to fit the 8×8 physical bitmap. This reduction results in smoothing the 'jagged edges' of light by using adjacent pixels to display calculated levels of light.

The library consists of several simple effects to control the particles. The effects include: move, pause, colour change, diminish brightness, and motion blur. These act as a base set of effects that can be used to build more complex animations. The effects can then be combined to work in series and/or parallel. For example, take a simple animation using the **Move** effect to move a particle from point A to point B. Add to this the **Dim** effect in parallel, which creates the animation of a particle moving from point A to point B while decreasing the brightness, so the particle is invisible by the time it reaches point B. Add to this the **Pause** effect to delay the start of the **Dim** until the point is perhaps halfway between point A and point B. The **Pause** and **Dim** effects would run together in series while the **Move** effect would run in parallel with the **Pause** and **Dim** effects. This is similar to the way a shooting star appears in the sky.

Creating fire is similar, but instead of a single particle, several are used simultaneously in a random manner. Each particle could be thought of as a spark or tongue of fire. Unlike a looping frame animation, this fire animation is generated programmatically to create a random fire.

Once the libraries are installed, the **fire_example** sketch can be accessed from the Arduino IDE using the menu option: File > Examples > ZContent_ ParticleFX. The code example above shows how the animation of a single fire particle is created with the **MotionBlur**, **Move**, **Pause**, and **Dim** effects.

Fire particles move vertically from the bottom of the screen to the top. The **MotionBlur** effect runs all the time, while the **Pause**, **Move**, and **Dim** effects run in series. The second parameter in the effect routines, **wait**, is a Boolean value that controls whether the effect should wait until it is completed before moving to the next effect. For example, a wait value of **true** will run this effect in series, while a wait value of **false** will run the current and next effects in parallel.

FROM SPARK TO FLAME

Eight particles are combined to create the fire simulation. So what about the horizontal motion? The fire animation uses only half of the 64

SAMPLE CODE

The full code is in the library example file which you can view at **hsmag.cc/tS7OHE**

```
fireEffects[i] = new particleEffectList();

#ifdef MOTIONBLUR

      fireEffects[i]->appendEffect(fireMotion
BlurEffect[i] = new
effectMotionBlur(fire[i], false, &firematrix,
&matrix, 6, 1.0));

#endif

// random wait to start flame

fireEffects[i]->appendEffect(firePauseEffect
[i] = new effectPause(fire[i], true,
random(0,8)));

// move flame up

fireEffects[i]-
>appendEffect(fireMoveEffect[i] = new
effectMove(fire[i], true, 8, 78, 6));

// dim flame

fireEffects[i]->appendEffect(fireDimEffect[i]
= new effectDim(fire[i], true, 0., 5));
```

GOING FURTHER

Control your flame remotely via Bluetooth using a Bluefruit-enabled Adafruit Feather. You can customise the example sketch to change the flame colour and/or turn it on or off, using the Bluefruit LE Connect app. You can find code examples for this on the Adafruit Learning System at **learn.adafruit.com**.

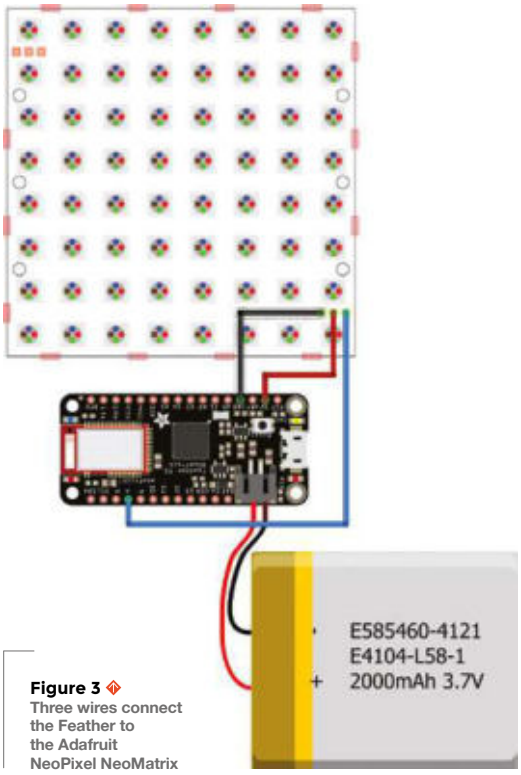
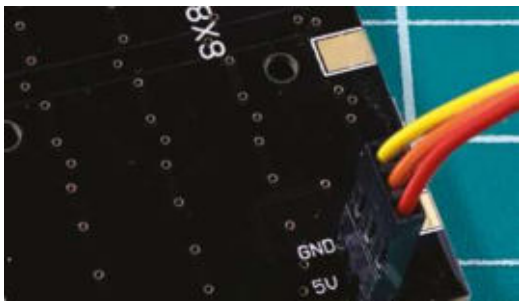


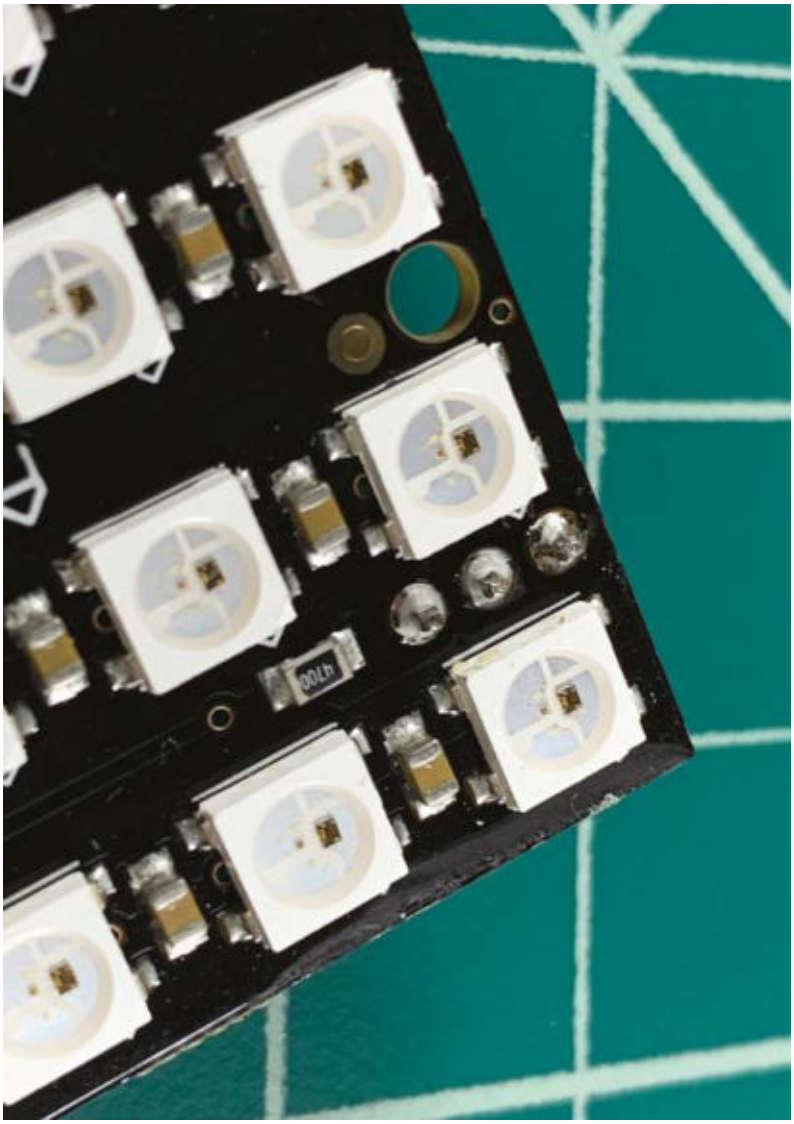
Figure 3
Three wires connect the Feather to the Adafruit NeoPixel NeoMatrix

NeoPixels on the PCB. The four centre columns are used exclusively since the simulated flame is rectangular rather than a square. One of the four centre columns is randomly selected at the start of the particle's motion, with the inner two columns being used 90% of the time, and the outer two columns 10% of the time. This simulates a fire with greater intensity in the centre, just like a real fire.

The animation frame rate defined in the code can be increased or decreased to alter the animation speed. The `framerate` defined in the code is the number of milliseconds between frames of the animation. A value of 40 is equivalent to .04 seconds per frame or 25 fps. A higher value for the `framerate` will slow down the animation (i.e. 250 is equivalent to 4 fps).

Now that you have created a fire in a jar, what other effects do you want to create for your jar? Try creating your own, or check the ZContent_ParticleFX library for new effect examples as they are released.





GETTING THE CODE

All the code used in this tutorial is open-source and available on GitHub. The ZContent_ParticleFX library contains the effect routines used to simulate fire, and the other libraries are used to display the animations on the NeoPixel NeoMatrix. These folders should be copied to your Arduino/libraries folder.

- ZContent_ParticleFX – github.com/ZContent/ZContent_ParticleFX
- Adafruit_GFX – github.com/adafruit/Adafruit-GFX-Library
- Adafruit_NeoMatrix – github.com/adafruit/Adafruit_NeoMatrix
- Adafruit_NeoPixel – github.com/adafruit/Adafruit_NeoPixel

Figure 4
Soldering the headers on the NeoPixel NeoMatrix requires steady hands as they are between two NeoPixels

Figure 5
Using headers allows one to easily connect and disconnect the NeoPixel NeoMatrix

TUTORIAL

Make organic-looking 3D prints

A little wood in your plastic makes all the difference



Ben Everard

@ben_everard

Ben loves cutting stuff, any stuff. There's no longer a shelf to store these tools on (it's now two shelves), and the door's in danger.

3D printers can make fantastically detailed parts in shapes that are hard to create using other automatic methods, but hobbyist-level machines can only work with plastic. If you prefer a different look and feel, there is a solution to this. Some filament manufacturers have filaments that are mostly plastic (usually polylactic acid, PLA), but also contain wood dust. This may account for up to 30% of the total volume, so it's still in the minority, but it's enough to give it a different look and feel to plastic.

Let's be honest straight away – no one is going to mistake this for wood, at least not up close. It's just not that realistic. However, it does have a different feel, texture, and even smell to plastic. The wood gives it a matt appearance, and the slight texture on the surface helps hide the layer lines. It's somewhere between wood and plastic.

THE PROOF IS IN THE PRINTING

This filament is usually made with PLA and, as such, it can be printed more or less exactly as PLA can be (see box opposite). The one caveat to this is nozzle size. It is possible to print woodfill on a standard 0.4mm nozzle; however, this is more or less at the edge of its abilities. It will be more prone to clogging – and bear in mind that not all woodfill is the same, and not all printers are the same.

We'd recommend switching to a 0.5mm or 0.6mm nozzle for woodfill. This will help reduce your chance of clogging significantly and, as a bonus, print much







faster. Since layer lines are less visible in woodfill (and people expect some 'grain' in the surface of wood anyway), it doesn't detract too much from the look of the prints.

Woodfill is a little more free-flowing than regular PLA, so can be prone to stringing. You can either fine-tune your retraction settings, or just deal with the stringing in post-processing. For simple models, a quick blast with the heat gun is probably easier than spending a few hours going through stringing tests with different parameters. But, if you'll be using it a lot, then it might be worth perfecting your settings.

One big thing to bear in mind about woodfill is that it's significantly weaker than regular PLA. As such, you probably shouldn't be using it for anything with any structural importance whatsoever. And even with purely decorative things, you might want to consider increasing the number of perimeters, or perhaps the infill density, to make sure it's strong enough.

We really like woodfill as it's a cost-effective way of making your prints look a little less plastic. It comes in a range of different colours and textures, so shop around to get something that appeals to you (many filament manufacturers will send you sample packs, so you can test a few out without committing to buying full kilos).

Above
We had a slightly blocked nozzle when running this print, but we like the effect

Left
At 0.4mm, there's no hiding the layers on this bead – even with woodfill – but a bit of sanding will remove them

TEMPERATURE CHANGES

One curious feature of woodfill is that it can char slightly at high temperatures, leaving it a different colour. You can utilise this to create a wood grain effect in your print by varying the temperature of your prints. There's a Python script, that also works as a plug-in for some versions of Cura, available on Thingiverse (**thingiverse.com/thing:49276**). Obviously, if you're heating something enough to char it, you really shouldn't be leaving it running unattended.

How much this changes the appearance of the print depends a lot on the particular filament. On some, it's no more than a slight tint, while on others, it's very noticeable.


CODE
THE
CLASSICS
VOLUME 1
Brimble
Crookes
Gillett





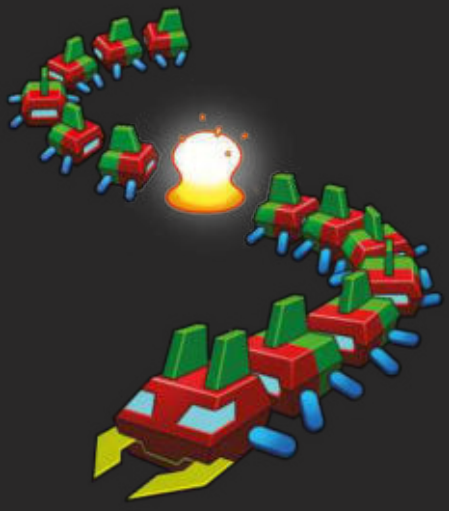
CODE THE CLASSICS
VOLUME 1

This stunning 224-page hardback book not only tells the stories of some of the seminal video games of the 1970s and 1980s, but shows you how to create your own games inspired by them using Python and Pygame Zero, following examples programmed by Raspberry Pi founder Eben Upton.





- ***Get game design tips and tricks from the masters***
- *Explore the code listing and find out how they work*
- ***Download and play game examples by Eben Upton***
- *Learn how to code your own games with Pygame Zero*



Available now **hsmag.cc/store**

The magic of magnets

How to use magnets in your shop and your projects



Gareth Branwyn

@garethb2

Gareth has been a lifelong practitioner (and chronicler) of DIY tech, media, and culture. He is the author of ten books, including *Tips and Tales from the Workshop*, and is a former editor for Boing Boing, Wired, and Make:.

When we think of useful tools in our shops and projects, magnets might not be the first thing that leaps to mind, but these little (and not so little) generators of magnetic fields can be a great way to temporarily hold things, pick things up, and clean up our metal messes. Here are some common, and not so common, uses for these often overlooked shop helpers.

Consider a magnetic sweeper If you have a shop or job-site that generates a lot of metal waste, consider getting a magnetic sweeper. They cost around $35, and have a release button for dumping the metal debris when you're done.



Below
We assume there's some science behind magnets, but they seem like magic to us

Make your own magnet dragnet If you only have a periodic need for picking up ferrous waste, you can make a temporary magnet picker by simply attaching a pole, or some ropes, to a magnetic kitchen knife rack (or another linear magnetic bar) and dragging it over the area you wish to clean.

Make a magnet stick or string Another handy all-purpose tool for picking up small metal items is a magnet glued to the end of a stick or string. Handy for grabbing hardware and metal shavings that have fallen behind benches and tools.

Creating a desktop magnet cleaner Having a chunk of magnet on hand is great for cleaning up work surfaces of any screws, electronics components, etc. If you place the magnet in a plastic bag, you can simply invert the bag to gather what you've caught.

Collecting debris from a drill-press or mill If you drill or mill metal, you can collect the filings as you generate them by placing a magnet (covered in a bag) near the tool head.

Make your tape-measure into a tool grabber Epoxy (or glue) a strong rare earth magnet to the end hook of your tape-measure. Then, if you drop a screwdriver or other light tool while on a ladder, you can extend the tape to pick it up.

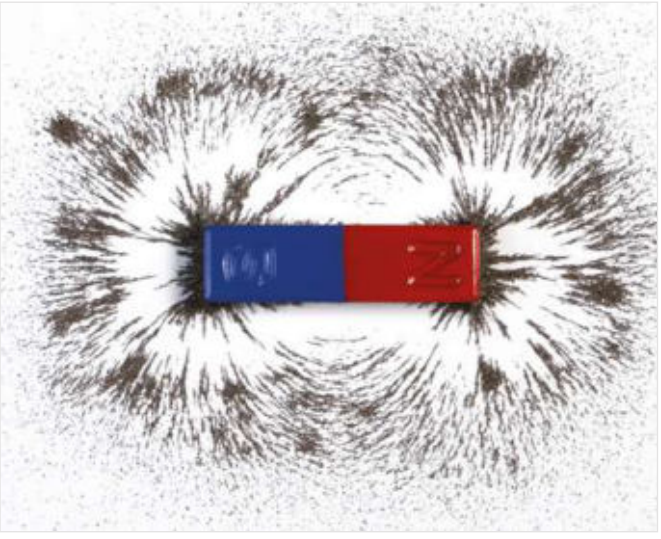
Add a tool and hardware catch to the top of your step ladder If you glue some magnets to the top of your ladder (or screw down a 'pot magnet' with holes in its centre), you



can create a place to securely hold light tools and hardware on the top step of your ladder.

Is that a magnet in your pocket? If you keep a magnet in your trouser pocket, you can use it to hold tools and fasteners on the outside of your pants.

Tool pouch magnet grab Toss a magnet into the tool pouch on your belt. Any hardware you toss in will be grabbed by the magnet.



Fetching metallic objects from pipes If you drop something small and metal down the drain, and your pipes are plastic, you can glue a magnet to a string, drop it down the pipe, and retrieve your object. Sadly, this won't work on that lost wedding ring.

Using magnets to find studs in a wall You don't need an electronic stud-finder tool to find the studs in a wall. Use a magnet to locate the screws in the dry wall, which will reveal the location of the studs.

Using magnets to exactly locate a pass-through hole If you want to be sure that you're cutting a hole in the exact spot where you need it to come through on the other side, use a pair of 0.75" diameter neodymium magnets to mark your target drill spot. Tape a magnet in place on one side of the board, find that spot on the opposite side using the second magnet, draw a circle around each magnet, and you're ready to drill.

Finding hidden nails, wire, brads, and screws If you're recycling lumber or pallet wood, there can easily be unseen hardware in the wood. To locate these hidden hazards, drag a rare earth magnet over the wood.

Chucks at the ready Place a magnet on the side of your drill-press or lathe to hold the chuck key when →

QUICK TIP

So, first off, what sorts of metals are magnets attracted to? Iron, nickel, and cobalt are the most common ferromagnetic materials. Steel is also ferromagnetic because it is an alloy of iron and other metals. Magnets for shops and projects come in a dizzying array of sizes and hold strengths. Rare earth (neodymium) magnets are the strongest. You can get them and other weaker types of magnets in various diameters and shapes. So-called 'refrigerator magnets' are weaker and good if you remove them a lot and need a light hold. For stronger jobs, use the rare earth type. 'Pot magnets' come in a steel housing (a 'pot'), are usually disc-shaped, and have a hole in the centre which makes them great for screwing to a surface.

Above
A magnetic bar keeping drill bits safe and sound

Left
If you ever get bored collecting swarf, you can re-create school physics classes and make magnetic field patterns

QUICK TIP

You can get bundles of refrigerator, pot, and rare earth magnets inexpensively online. They come in all diameters of discs, blocks (squares and rectangles), and bars. For instance, you can get 100 fridge magnets for under £10, or 15 32mm neodymium magnets for £14.

it's not being used. You can also use magnets stuck (or glued or taped) to your machinery to hold adjustment wrenches and other needed tools. You can also attach a magnetised, retractable key chain, so your chuck key is always attached to your drill.

Magnetic kitchen-knife holders Magnetic knife rails can be placed in all sorts of locations around the shop to hold tools, bits, hardware, all sorts of things – attach them to walls, inside cabinet doors, on the edges of workbenches, just about anywhere you need to grab a tool. You can find these holders online for about £5 each.

Store bits with what they drive For screws and other fasteners, especially speciality bits, glue a small magnet to the lid of the screw container, and store the bit with the fasteners.

Make your own magnet trays A tuna can with a magnet on the bottom will hold screws and other hardware securely until you need them.

> A tuna can with a magnet on the bottom will hold **screws and other hardware securely until you need them**

Above
Don't chase metal parts around with your fingertips: hold them steady with a magnet

Below
Knife holders are designed for kitchens, but can be just as useful as tool holders in the workshop





Mechanic's parts trays You can find magnetic mechanic's parts trays online for £10 to £20. They come in round, square, and rectangular configurations and various sizes. These are great to use on top of tool-boxes because the contents stay in the tray, even if you lift the lid.

Lathe tool holder A big magnet on top of a lathe housing will stop Allen keys, tool bits, drills, etc. from falling off due to vibrations.

Keep metal filings out of shop fluids Place a magnet in the bottom of cans of cutting fluid and other lubricants you use around metalworking machines. When you place the brush back in the can, any filings caught on the brush will be pulled to the magnet.

Hold bolts in place with magnets If you have a wing bolt or other fastener that frequently has to be unscrewed (e.g. on a camera mount), fasten a magnet next to it so that you don't have to hold it in place every time you unscrew what's attached to it.

Use a bar magnet as a temporary stop block To create a temporary stop on a table saw, chop saw, or other shop tools with a metal table, place a shop magnet (also called a holding magnet or fishing magnet) where you want the stop to be. This is great for making repetitive cuts without having to deal with clamps and wooden stop blocks.

Holding access panels in place If you have access panels around your shop, house, or on tools, consider using magnets to hold the panel in place. A simple hole in the panel is all you need to remove it.

Door-stops Sink strong magnets into the edges of double doors to stop them from coming open in a

breeze without needing a physical latch. You can use this technique to create hidden latches and clasps on doors, chests, small boxes, etc.

Probes and other instrument holders Old hard drive magnets can be stuck to the side of your computer to hold the probes of a PC-based oscilloscope or a borescope.

Magnetise a tool You can temporally magnetise a tool by attaching a small magnet to its shaft. This will allow hands-free holding of screws and fasteners. To more semi-permanently magnetise the actual driver, check out this video: **youtu.be/_E2vLos39Yg**.

Magnetise Allen keys If you use set screws, it's useful to magnetise your Allen keys, particularly if you use ball-head wrenches.

Reusing magnets Magnets can be harvested from old consumer electronics, such as stereo speakers and hard drives. You can then mount these on a wall or tool board and use them to hold your tools. □

Below
One of the most useful magnet tricks – finding which way is north





Above
Magnets are useful for holding things of all sizes, from screws to cars

MAKING A MAGNETIC GLUING JIG

If you do a lot of gluing of small, lightweight parts (e.g. styrene plastic, balsa wood, etc.), consider using a magnetic gluing jig. These are popular among scale modellers, doll-house builders, and model train enthusiasts. You can buy these jigs with magnet sets for around £25 to £40, but you can easily make your own for half that price. All you need is a piece of sheet steel (you can get a 12"× 12" piece online for under £10), some strip wood to create the sides (you can use paint stirring sticks if they're long enough). You will also need some holding magnets (you can get 30 10 × 10 mm magnets for under £10).

Create a frame around the sheet steel with the strip wood. Use bracing underneath as needed to create a sturdy and square frame. It's critically important that the frame is perfectly square.

You can now use the magnets to hold workpieces in place, against the frame or in the middle of the steel sheet, as you do your glue-ups. If you want to build models and other structures directly on top of plans, you can forgo (or remove) the frame and use temporary spray mount to attach the plans directly to the steel sheet. From there, you can build your structure using the magnets as your clamps.

One version of a right-angle jig for gluing styrene models can be seen on YouTube at TinkerTerrain: **youtu.be/bG8saJ_GzpQ**.



THANKS
Michael Colombo, Andrew Lewis, Becky Stern, Tom Haney, Bob Knetzger, Tim Sway, Richard Gould, Steve Roberts

Create low-power, clock-controlled devices

Consume close to zero power when your sensor isn't sensing



Rob Miles

@robmiles

Rob Miles has been playing with hardware and software since almost before there was hardware and software. You can find out more about his so-called life at **robmiles.com**

Above A sleeping Raspberry Pi Zero apparently consuming no current!

Do you want to make a sensor with a battery life you can measure in days rather than hours? Even if it contains a power-hungry device like a Raspberry Pi? By cunning use of a real-time clock module, you can make something that wakes up, does its thing, and then goes back to sleep. While asleep, the sensor will sip a tiny amount of current, making it possible to remotely monitor the temperature of your prize marrow in the greenhouse for days on end from a single battery. Read on to find out how to do it.

REAL GOOD TIME CLOCK

We'll be using a DS3231 real-time clock which is sold in a module, complete with a battery. The DS3231 contains two alarms and can produce a trigger signal to control a power switch. To keep our software simple, we are going to implement an interval timer, but there is nothing to stop you developing software that turns on your hardware on particular days of the week or days in the month. The DS3231 is controlled using I^2C, which means it can be used with lots of devices.

MOSFET ACCOMPLI

The power to our Raspberry Pi Zero is controlled via a P-channel MOSFET device operating as a switch. The 3.3V output from Raspberry Pi is used to power the DS3231 and our BME280 sensor. The gate on the MOSFET is connected via a resistor network to the SQW output from the DS3231.

You can think of a MOSFET as a kind of switch. It has a source pin (where we supply power), a drain pin (which is the output the MOSFET controls), and a gate pin. If we change the voltage on the gate pin, this will control whether the MOSFET conducts or not.

We use a P-channel MOSFET to switch the power because the gate voltage must be pulled down to cause the MOSFET to conduct, and that is how P-channel devices function.

MOSFET devices are all about voltage. Specifically, when the voltage difference between the source and the gate pin reaches a particular value, called the threshold voltage, the MOSFET will turn on. The threshold voltage is expressed as a negative value because the voltage on the gate must be lower than the voltage on the source. The MOSFET that we're using turns on at a threshold voltage of around -3.7 volts and off at a voltage of -1.75 volts.

The SQW signal from the DS3231 is controlled by a transistor which is acting as a switch connected to ground inside the DS3231. When the alarm is triggered, this transistor is turned on, connecting the SQW pin to ground. The diagram in **Figure 1** (overleaf) shows how this works.

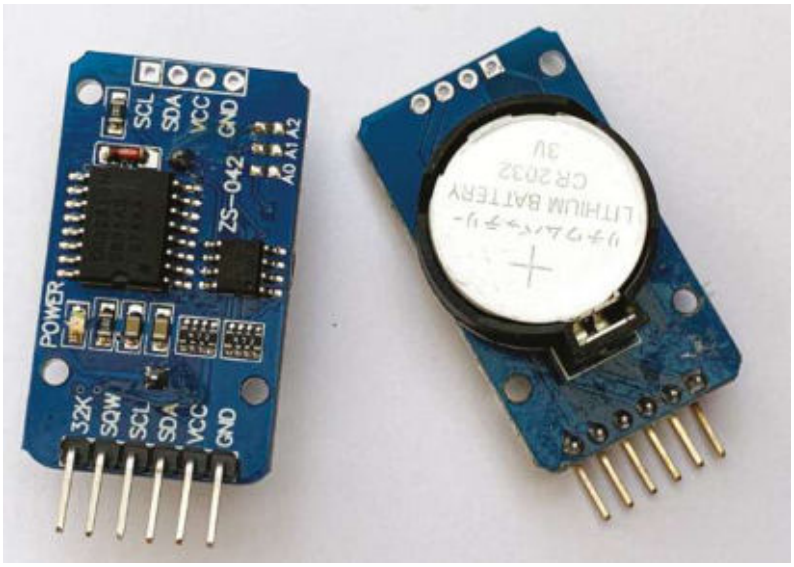
> The program does whatever it needs to do. This is **the action that you want to perform at regular intervals**

The resistors R1 and R2 are linked to the supply voltage at one end and the SQW pin and the MOSFET gate on the other. When SQW is turned off the voltage on the MOSFET gate is pulled high by the resistors, so the MOSFET turns off. When SQW is turned on, it pulls the voltage on the MOSFET gate down, turning it on.

Unfortunately, current leaking through R1 and R2 to the DN3231 means that we are not going to get zero current consumption when the MOSFET is turned off, but it is much less than 1 milliamp.

POWER CONTROL

Now that we have our hardware built, we can get some code running to control the power. The DS3231 is connected to Raspberry Pi using I²C. Before you start, you must enable I²C on your Raspberry Pi using the raspi-config tool. Use `sudo raspi-config` and select Interfacing Options. Next, you need to make sure that you have all the I²C libraries installed by issuing this command at a Raspberry Pi console:



Above
You can pick up one of these modules from lots of suppliers. Make sure that you get one with the SQW connection, as that provides the alarm signal

```
sudo apt-get install python3-smbus python3-dev
i2c-tools
```

The sequence of operation of our sensor is as follows:

1. The program does whatever it needs to do. This is the action that you want to perform at regular intervals. That may be to read a sensor and send the data onto the network, or write it to a local SD card or USB memory key. It could be to read something and update an e-ink display. You can use your imagination here.
2. The program then sets an alarm in the DS3231 at a point in the future, when it wants the power to come back on.
3. Finally, the program acknowledges the alarm in the DS3231, causing the SQW alarm output to change state and turn off the power.

CLOCK SETTING

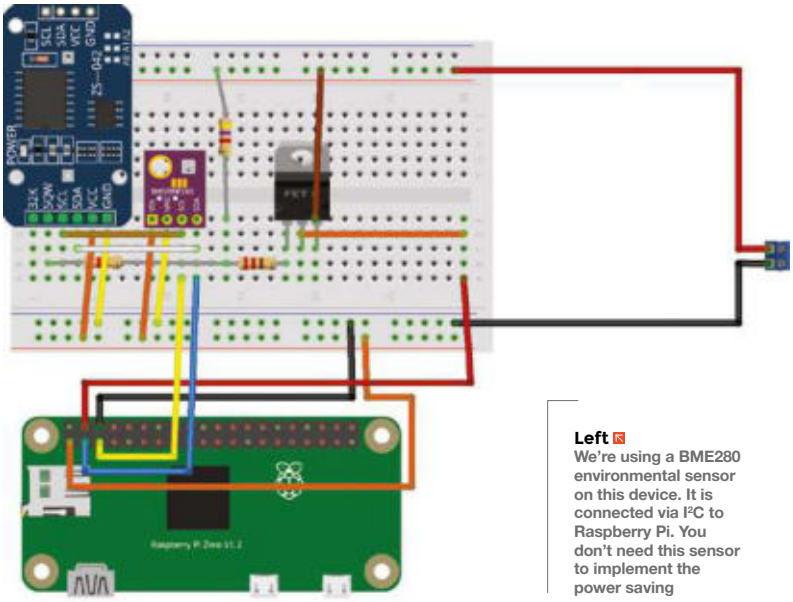
The program below only uses a fraction of the capabilities of the DS3231 device. It creates an →

DEBUGGING

It can be hard to debug your programs when your program keeps turning off the power to your device. To get around this, you can add a switch that connects SQW to ground. When you close the switch, it will turn on the MOSFET and power up your device.

YOU'LL NEED

- **DS3231 powered real-time clock module with battery backup**
 Make sure it has a battery holder and an INT/SQW output pin
- **P-channel MOSFET**
 The IRF9540N works well
- **Three resistors: 2.2 kΩ, 4.7 kΩ, and 220 Ω**
- **A device you want to control**
 This can be a PIC, Arduino, ESP8266, ESP32, or Raspberry Pi. My software is written in Python and works in MicroPython or on Raspberry Pi, but you can find DS3231 driver software for lots of devices
- **Sensor you want to use**
 We're using a BME280 to get air temperature, pressure, and humidity
- **Breadboard or prototype board to build up the circuit**



Left
We're using a BME280 environmental sensor on this device. It is connected via I²C to Raspberry Pi. You don't need this sensor to implement the power saving

QUICK TIP

It is called the SQW pin because the DS3231 can be configured to produce a square wave signal on this pin which can be used by circuits needing a regular clock signal.

interval timer that can time hours, minutes, and seconds. Each time the program runs, the clock is set to zero, and the alarm is configured to trigger when the target time is reached.

Put the program into a file called **SensorAction.py** on your Raspberry Pi, and put the code that you want to run into the section indicated.

```
import smbus

bus=smbus.SMBus(1)

DS3231=0x68

SECONDS_REG=0x00
ALARM1_SECONDS_REG=0x07

CONTROL_REG=0x0E
STATUS_REG=0x0F

def int_to_bcd(x):
    return int(str(x)[-2:], 0x10)
```

MICROCONTROLLERS

You can also use CircuitPython or MicroPython to interact with the DS3231 over I²C. The I²C commands are slightly different, but the fundamentals are the same. Look in the GitHub repository for this article for more details.

POWER DOWN

The SensorAction program turns off your Raspberry Pi without shutting it down properly, which is something your mother probably told you never to do. The good news is that in extensive testing, we've not experienced any problems with this. However, if you want to make your Raspberry Pi totally safe in this situation, you should make its file system 'read-only', which means that it never changes during operation and therefore can't be damaged by untimely power cuts. There are some good instructions from Adafruit here: **hsmag.cc/UPgJSZ**.

Note: making the operating system file store read-only does not prevent you creating a data logging application, but you would have to log the data to an external USB key or SD card and then dismount the storage device before killing the power.

If you are using a different device, such as an ESP8266 or an Arduino, you don't need to worry about this as the software in them is inherently read-only.

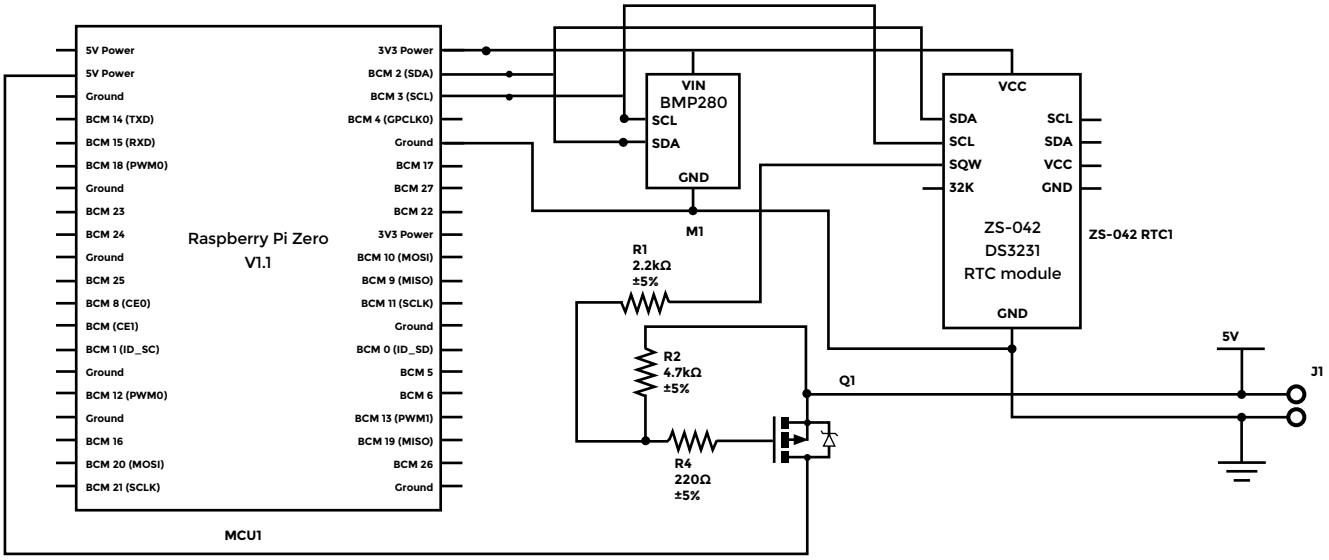
```
def write_time_to_clock(pos, hours, minutes,
seconds):
    bus.write_byte_data(DS3231, pos, int_to_
bcd(seconds))
    bus.write_byte_data(DS3231, pos+1, int_to_
bcd(minutes))
    bus.write_byte_data(DS3231, pos+2, int_to_
bcd(hours))

def set_alarm1_mask_bits(bits):
    pos=ALARM1_SECONDS_REG
    for bit in reversed(bits):
        reg = bus.read_byte_data(DS3231, pos)
        if bit:
            reg = reg|0x80
        else:
            reg = reg&0x7F
        bus.write_byte_data(DS3231, pos, reg)
        pos = pos+1

def enable_alarm1():
    reg = bus.read_byte_data(DS3231, CONTROL_REG)
    bus.write_byte_data(DS3231, CONTROL_REG,
reg|0x05)

def clear_alarm1_flag():
    reg = bus.read_byte_data(DS3231, STATUS_REG)
    bus.write_byte_data(DS3231, STATUS_REG,
reg&0xFE)

def check_alarm1_triggered():
    return bus.read_byte_data(DS3231, STATUS_
```




```
REG)&0x01 != 0

def set_timer(hours, minutes, seconds):
    # zero the clock
    write_time_to_clock(SECONDS_REG, 0, 0, 0)
    # set the alarm
    write_time_to_clock(ALARM1_SECONDS_REG,
hours, minutes, seconds)
    # set the alarm to match hours minutes and
seconds
    # need to set some flags
    set_alarm1_mask_bits((True, False, False,
False))
    enable_alarm1()
    clear_alarm1_flag()
#
# Your sensor behaviour goes here
#
set_timer(1,30,0)
```

The `set_timer` function is called to set the timer and clear the alarm flag. This resets the alarm signal and powers off the sensor. The example above will cause the sensor to shut down for 1 hour 30 minutes.

ALWAYS RUNNING

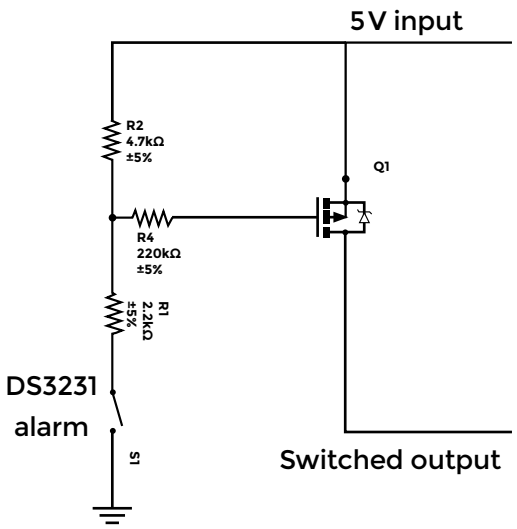
To get the program to run when the Raspberry Pi boots, use the Nano editor to add a line at the end of the **rc.local** file that runs your program.

```
sudo nano /etc/rc.local
```

Use the line above at the command prompt to start editing the **rc.local** file and add the following line at the end of the file:

```
python3 /home/pi/SensorAction.py &
```

This statement runs Python 3, opens the **SensorAction.py** file, and runs it. Don't forget the ampersand (&) at the end of the command: this starts your program as a separate process, allowing the boot to complete. Now, when Raspberry Pi boots up, it will run your program and then shut itself down. You can find a full sample application on the GitHub pages for this project (**hsmag.cc/Yx7q6t**). It logs air temperature, pressure, and humidity to an MQTT endpoint at regular intervals. Now, go and start tracking that marrow temperature!



Above
You can use any other microcontroller that implements I²C

Figure 1
The SQW output from the DS3231 will pull the gate of the MOSFET low to turn on the power to Raspberry Pi

QUICK TIP

Don't try to use an N-channel MOSFET. They switch on when their input gate is pulled high, which is no good to us.



Build your own vacuum former

Use an Arduino and a thermocouple to shape plastic

Vacuum forming is a relatively simple process that lets you melt and shape plastic over a form using suction. Although it isn't a complicated technique to use, it's often neglected because the equipment needed to melt the plastic and pull a vacuum isn't something most people have access to. In this project, you'll see how to put your own vacuum forming machine together using common parts, and control everything using an Arduino.

VACUUM FORMERS SUCK!

The main parts of a vacuum forming machine are the frame, the heating element, and the vacuum plate. The frame holds a plastic sheet in place while it's being heated and formed; the heating element is a simple grill; and the vacuum plate is a mesh plate that connects to a vacuum source. The vacuum source for this project will be a shop-vac (or any vacuum cleaner with a hose), and the vacuum plate will be made from a few household items.

Begin by removing the grill from the disposable barbecue, and cutting it to the same size as one of the large IKEA roasting trays.



Dr Andrew Lewis

@monkeysailor

Dr Andrew Lewis is the owner of **Shedlandia.com**, a restorer of old tools, a fabricator for hire, a research scientist, and a founder member of the Guild of Makers.



Above
You'll need to make a hole inside of a large IKEA oven tray to accommodate the hose from your vacuum. The easiest way to do this is to cut the hole with a jeweller's fret-saw, and use the 3D-printed hose fitting with 3mm bolts to finish the edge. If you don't have access to a 3D printer, you can just cut the hole in the metal, and make a plastic gasket to fit on either side

This roasting tray will be the base of your vacuum forming machine. Attach the 4040 extrusion vertically to the mid-point at the back of the tray by drilling through the tray and fixing it in place with two T-nuts and 8mm bolts with washers to reinforce the thin metal of the tray. Use a PTFE block as a spacer between the tray and the 4040 extrusion to overcome the lip on the tray, drilling through the block using the existing holes in the tray as a guide.

Make a rectangular frame from aluminium angle that fits around the top of the roasting tray, with a small cut-out at the back to allow clearance for the 4040 extrusion, and another cut-out to allow for the vacuum hose. Glue the grill from the barbecue inside the frame using silicone rubber, and then fit it down onto the top of the roasting tray with more silicone.

You will need to make a U-shaped bracket from steel to hold the trays under the top grill. The position of this bracket will affect how hot the plastic gets, so some experimentation may be necessary to get the best results. Fitting the bracket to the extrusion with an M8 bolt and T-nut makes adjusting the height easy.

SAFETY

This project uses high temperatures and mains voltages. The unit should never be left unattended when running as there is a fire risk (and also, it's a manual machine so needs user input). Make sure you take proper precautions to avoid burning yourself when using it – hot plastic is particularly unpleasant.

If you are unsure about working with mains voltage, make sure you seek advice from someone who is suitably experienced before attempting this project. Take particular care to make sure that the unit is properly grounded, as this may protect you in the unlikely event of a short circuit.

GRILLED CHEESE, ANYONE?

The top grill is more complicated to make, because it needs to be able to deal with high temperatures. That means no soldering or plastic wire sheathing. We chose to use GU10 halogen bulbs as the source of heat, because they are easily available and we already had all of the necessary fittings. Begin by marking out and drilling the holes for the GU10 fittings on the bottom of one of the smaller IKEA roasting trays. →

Below
This author used aluminium brazing to fix the aluminium frame together, but it should be possible to use nuts and bolts if you don't want to resort to a propane torch



YOU'LL NEED

- 1 m 4040 aluminium extrusion
- 8 × M8 sliding nut for 4040 extrusion
- 12 × M8 30 mm cap head bolt
- 4 × M8 dome nut
- 8 × 9.5 mm × 19 mm compression springs (or similar)
- 3 × 30 mm M3 pan head screws
- 3 × 50 mm M3 pan head screws
- 8 × M3 nuts
- 1 m 25 mm steel angle
- 1 m 23.5 mm × 43.5 mm unequal aluminium angle (or steel angle)
- 2 × PTFE block, approx 40 mm × 50 mm
- Family-size disposable barbecue
- 1 m 1.2 mm piano wire
- 8 × GU10 sockets
- 8 × GU10 50 W halogen bulbs
- 2 × large baking trays
- 2 × IKEA KONCIS roasting trays with rack (40 cm × 32 cm)
- 2 × IKEA KONCIS roasting trays (34 cm × 24 cm)
- 2 × ASDA Home 46 cm oven tray (or similar)
- Arduino Uno (or similar)



Above
The smaller tray with the light fittings hangs inside the larger roasting tray, using four 8mm bolts held in place with nuts, washers, and springs, but you'll need to drill a cable hole in the larger tray and connect the silicone power wires to the ceramic connector block before you bolt them together. You can see the hanger bolt in this image with the bottom spring, bolt, top spring, and washer fitted



Above
The frame is very easy to make, and just involves cutting a suitably sized hole through two of the oven trays, and adding a bead of silicone rubber around the bottom tray to hold the vacuum once it's placed onto the vacuum plate. The plastic is sandwiched between the trays when it's heated, so that it can be manoeuvred into position on the vacuum plate. Hold the trays together (wearing thermally insulated gloves, obviously) with some bolts or clamps on the outside, to stop the plastic from sliding out of place

YOU'LL NEED CONT...

- **ELEGOO 4-channel relay board with optocoupler**
- **1 × 2 A RobotDyn AC light dimmer module**
- **240 mm × 190 mm × 90 mm IP56 junction box**
- **4 × DPST rocker switch**
- **1 × NO momentary 12 mm button**
- **0.91-inch I²C OLED**
- **10 kΩ panel mount potentiometer**
- **8 × 5 mm red LED**
- **AD8495 thermocouple amplifier break out**
- **K-type thermocouple**
- **1 m 10 mm silicone tubing**
- **4 m (approx) of high-temperature silicone cable** (at least 20AWG)
- **4 × 2-way ceramic terminal blocks**
- **1 × 10 A stop/start switch**
- **3D-printed panel and 3D-printed hose fitting**

The fitting method will vary depending on the exact type of GU10 fitting you have, but we drilled a hole large enough for the fitting to pass through, and held it in place with a spring bent from piano wire. This method held the fitting in place securely, and required no extra drilling. The wires from the fitting are already designed to take high temperatures, and will be wired together in pairs with each switch on the control panel connecting to two of the lights. Wire the pairs of lights together using four ceramic connector blocks, and mount the blocks to the tray using 3mm nuts and bolts. The other large IKEA roasting tray is used as a cowl to hide the wiring away and provide protection against the hot metal. The outer tray fits to the 4040 extrusion in the same way as the tin on the base, using a PTFE spacer and two 8mm bolts. You also need to drill a 10mm hole to allow the wires.

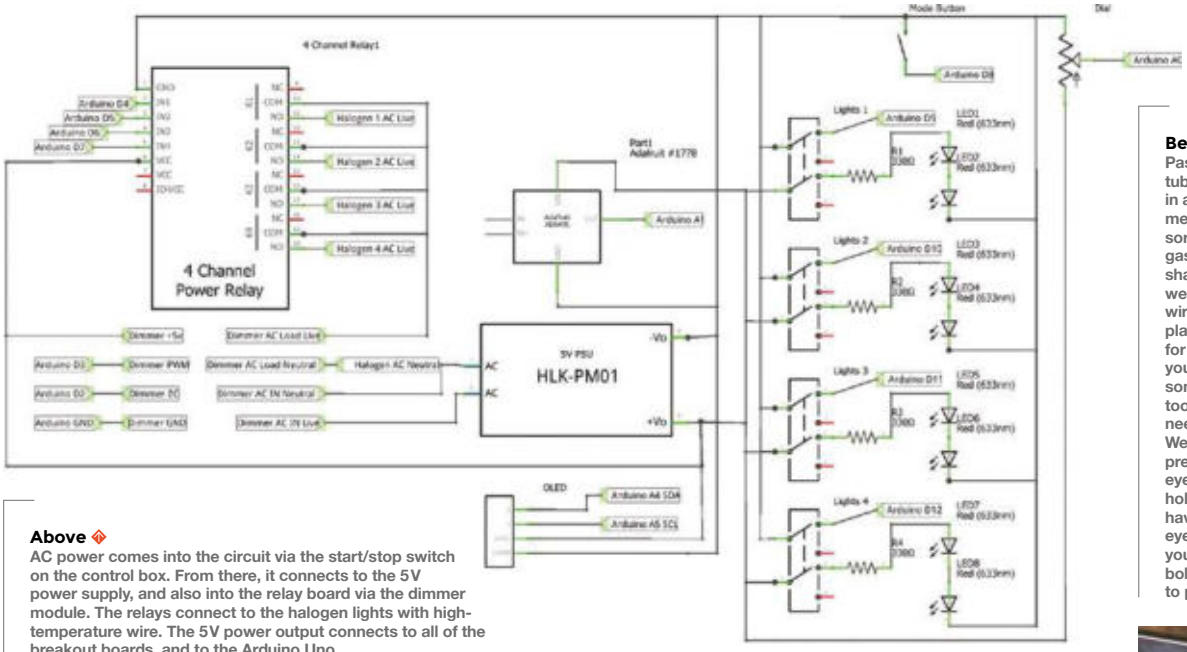
> "The controller will turn the lights off and **flash the word 'OVERHEAT' on the OLED display**"

THAT'S HOT STUFF!

The control box for this project is mounted at the top of the 4040 extrusion and held in place using T-nuts, washers, and M8 bolts. The front panel has a dial, a button, four switches, an LED display to show which lights are switched on, and an OLED display to provide text feedback. There are two modes of operation (smart and simple), and you can toggle between them using the push-button next to the OLED display. In simple mode, the OLED display will display a number in percent, which reflects the brightness of the bulbs. The potentiometer acts as a simple dimmer, and the four switches turn banks of halogen lights on and off. The smart mode uses a thermocouple to detect and maintain the temperature of the lights by increasing or decreasing their brightness automatically. In smart mode, the OLED displays the current temperature, and the desired target temperature. The potentiometer allows you to set the target temperature. If the thermocouple detects too high a temperature, the controller will turn the lights off and flash the word 'OVERHEAT' on the OLED display until the unit has cooled down.

EXTRA USES

The smart mode functionality means that you can use the vacuum former grill to reflow solder. To reach the temperatures needed to reflow, you need to make a special frame with a small IKEA roasting tray mounted on an ASDA tray so that it sits directly underneath the grill, minimising heat loss. You'll also need to put a small hole in the side of the small IKEA roasting tray to pass the thermocouple inside. The item to be reflowed sits inside this tray, which essentially makes the grill into a small oven.



Above
AC power comes into the circuit via the start/stop switch on the control box. From there, it connects to the 5V power supply, and also into the relay board via the dimmer module. The relays connect to the halogen lights with high-temperature wire. The 5V power output connects to all of the breakout boards, and to the Arduino Uno

Position all of your components inside the box and connect them as shown in the circuit diagram for the project, then flash the Arduino with the project code. Remember that you're working with high voltages, so make sure you route every cable carefully, and double-check every connection. Pay particular attention to the earth cables, and don't plug anything in until you've double-checked that the lid is on the control box and there is no continuity between the AC power and the chassis. When you're absolutely sure it's safe, fit the GU10 bulbs into the grill, and plug the unit in to test everything is working as expected.



Right
Six silicone-coated wires connect the halogen lights to the main control box, using silicone tubing as a conduit to carry the wires. Four of the wires are AC positive and go to the relay board, while the other two are AC negative and go directly to the dimmer unit's load negative

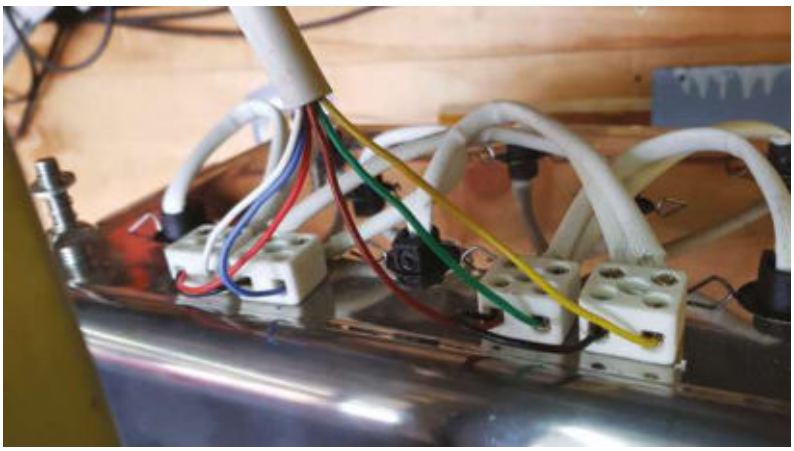
TESTING TEMPERATURE

Thermistor amplifiers need a stable, accurate, source voltage. To get accurate readings, you need to make sure you set the AREF value in the thermocouple setup section of the code to reflect the actual voltage being supplied to the amplifier breakout. Some boards, like the MAX31855 and the MAX6675, provide a digital SPI or I²C output from a thermocouple, and these are less prone to slight variations in input voltage. The thermocouple amplifier board used in this project provides an analogue output, which only requires a single pin connection to the Arduino.

Below
Passing cables and tubes through a hole in a piece of sheet metal needs some sort of protective gasket to stop the sharp metal from wearing through the wire. Normally, a plastic gasket is fine for this, but when you're working with something that will get too hot for plastic, you need an alternative. We used an eyelet press and brass eyelets to finish the holes, but if you don't have access to an eyelet press or punch, you could use a brass bolt or plumbing fitting to protect the wire



Below
Six silicone-coated wires connect the halogen lights to the main control box, using silicone tubing as a conduit to carry the wires. Four of the wires are AC positive and go to the relay board, while the other two are AC negative and go directly to the dimmer unit's load negative



BOOK OF MAKING

VOLUME 2



ONLY £10

WHSmith

BARNES&NOBLE

Available on the App Store

Google Play

THE BEST PROJECTS FROM

HACKSPACE MAGAZINE

THE ULTIMATE SKILLS, TRICKS, AND MAKES

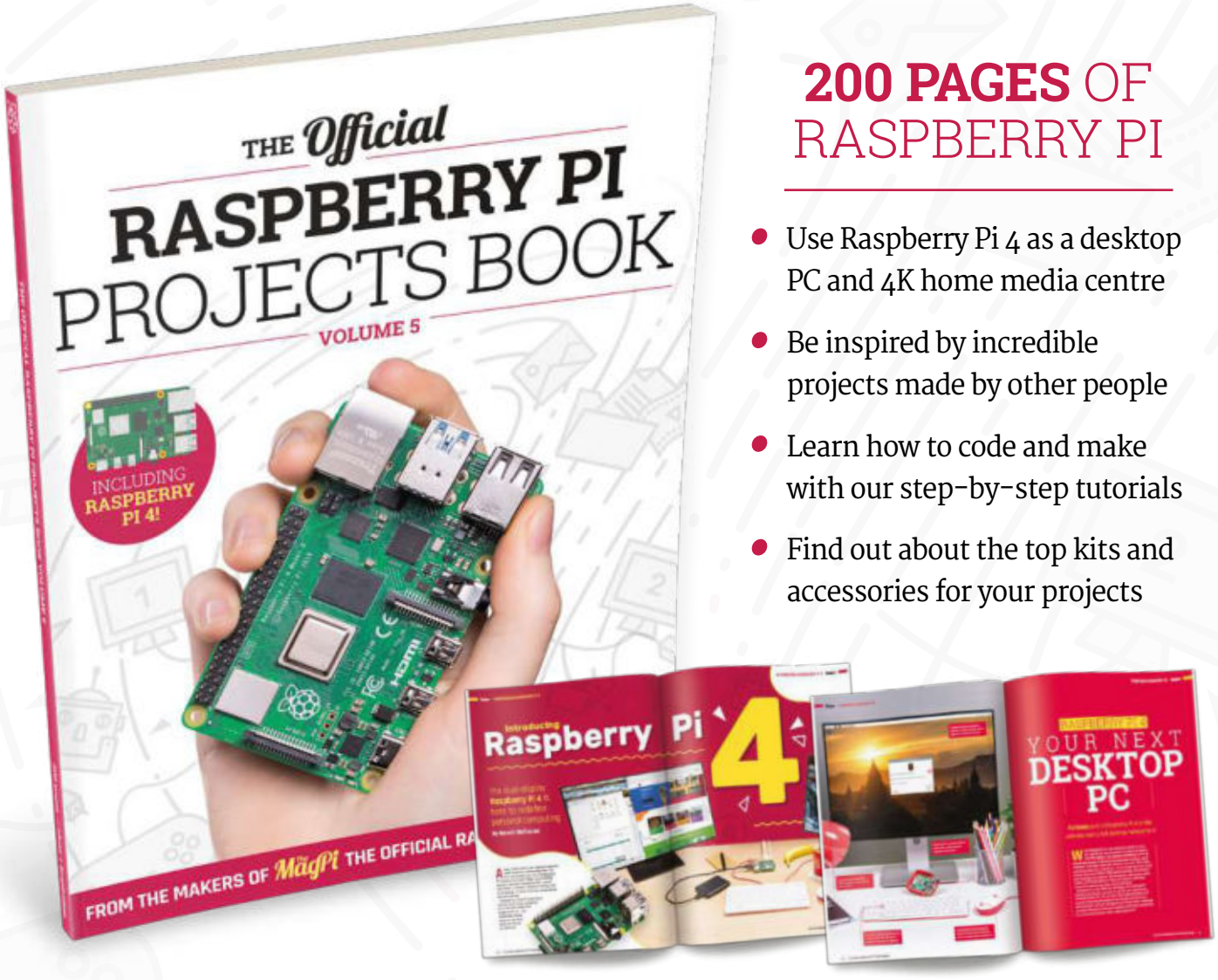
AVAILABLE NOW

hsmag.cc/store

FROM THE MAKERS OF **HackSpace** MAGAZINE

THE *Official* RASPBERRY PI PROJECTS BOOK

VOLUME 5



200 PAGES OF RASPBERRY PI

- Use Raspberry Pi 4 as a desktop PC and 4K home media centre
- Be inspired by incredible projects made by other people
- Learn how to code and make with our step-by-step tutorials
- Find out about the top kits and accessories for your projects

Buy online: magpi.cc/store



Electronic affirmation mirror

Build a circuit to cast your reflection in a positive light



Becky Stern

@bekathwia

Becky Stern is a lifelong maker living in NYC. She's published hundreds of DIY tutorials and videos about everything from microcontrollers to knitting. Find her on YouTube and **beckystern.com**

When you look in the mirror, who couldn't use a few words of encouragement? Build a display inside a mirror to scroll custom affirmations you can read over your own reflection. This polished project comes together easily with a store-bought shadow box frame and some see-through mirror material. You can build a simple version with a basic Arduino or compatible, or level it up using a NodeMCU (ESP8266 WiFi board) in order to add more positive quotes to the mirror over the internet.

To get started, first assemble the alphanumeric displays by soldering them to their I²C backpacks. The backpack is a circuit board containing a chip to control each of the LED segments through the I²C data protocol, which uses just two data pins to control the large number of LEDs. The delicate segment display pins go through the holes on the board; then you can lay the display down on its face and solder the pins at the back. Be careful to line up the decimal dot LED with the dot on the board's silkscreen to avoid soldering it in upside down! It's smart to solder two diagonal corner pins first, so you can make any adjustments to the seating before everything's locked in place. Components with multiple pins are exceedingly difficult to de-solder later. After all the leads are soldered, clip the protruding wire leads with a pair of flush snips.

Next, you should solder the included header pins into the labelled row of holes at the top of the display. It's easier to get everything lined up straight if you do this part of the assembly with the header pins sticking into a solderless breadboard (**Figure 1**).

You can chain these I²C devices together, so long as they have different addresses. You should change the address of two of the displays by soldering to bridge a set or two of the pairs of pads on the back of

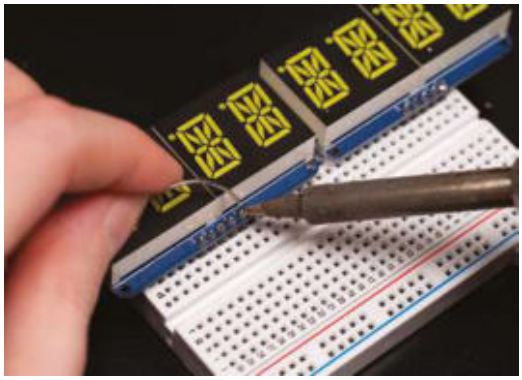


Figure 1
It's helpful to start by soldering just one header pin, which you can easily remelt to make any adjustments if the board and pins aren't perfectly perpendicular at first

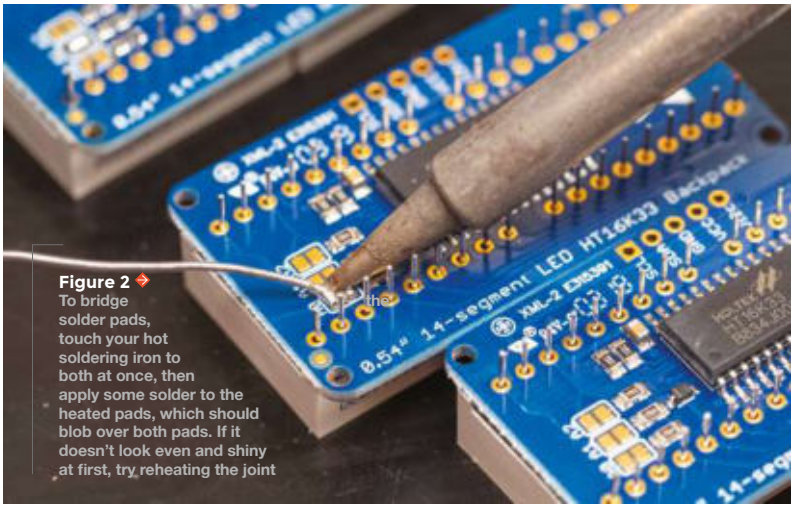


Figure 2
To bridge solder pads, touch your hot soldering iron to both at once, then apply some solder to the heated pads, which should blob over both pads. If it doesn't look even and shiny at first, try reheating the joint

the board. For instance, bridge the pads marked A0 on one display, and the pads marked A1 on another, leaving the third one at its default address (**Figure 2**). You don't have to use three displays, though! It still looks good and readable with just two, albeit harder to photograph a meaningful phrase since so few letters appear at once.

TEST IT, NO EXCUSES!

Test your displays with a solderless breadboard before assembling them into your project (**Figure 3**). That way, if something goes wrong, you won't have to spend a bunch of time trying to test multiple variables at once or have to waste time undoing and redoing work unnecessarily. You can use the alphanumeric display's Arduino library example code to test each screen individually. The simple code available with this project includes a sketch to scroll a test across three displays at once, which helps with identifying and confirming the correct address for each display.

```
// label the displays with their i2c addresses
struct {
  uint8_t           addr;         // I2C address
  Adafruit_AlphaNum4 alpha4;
} disp[] = {
  { 0x70, Adafruit_AlphaNum4() },  // left-hand
display
  { 0x72, Adafruit_AlphaNum4() },  // middle
display
  { 0x71, Adafruit_AlphaNum4() },  // right-hand
display
};
```

Look for this block of code at the beginning of the quadalphanum-multi example. If your displays aren't scrolling in order, try swapping the addresses (0x70, 0x72, 0x71) and re-uploading your code. If you soldered a different configuration of address pins than pictured, the other addresses you might try are 0x73, 0x74, 0x75, 0x76, and 0x77. The address is calculated as a sum of 0x70 and the bridged address pins, each of which has its own value (A0 = 1, A1 = 2, A3 = 4).

(SOLDER) SMOKE AND MIRRORS

Now it's time to lay out the final circuit on a big perfboard. Align it to one of the bottom corners of your shadow box frame, then place your displays so that they are centred in the frame from right to left, →

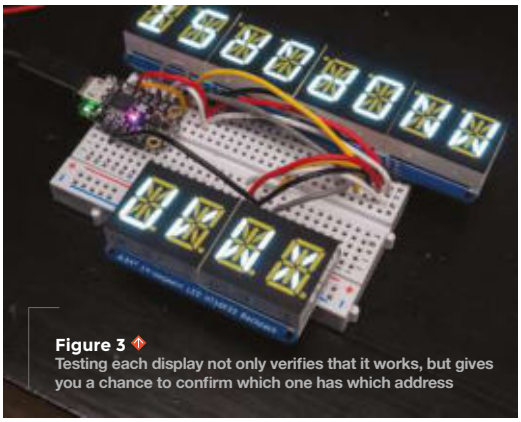


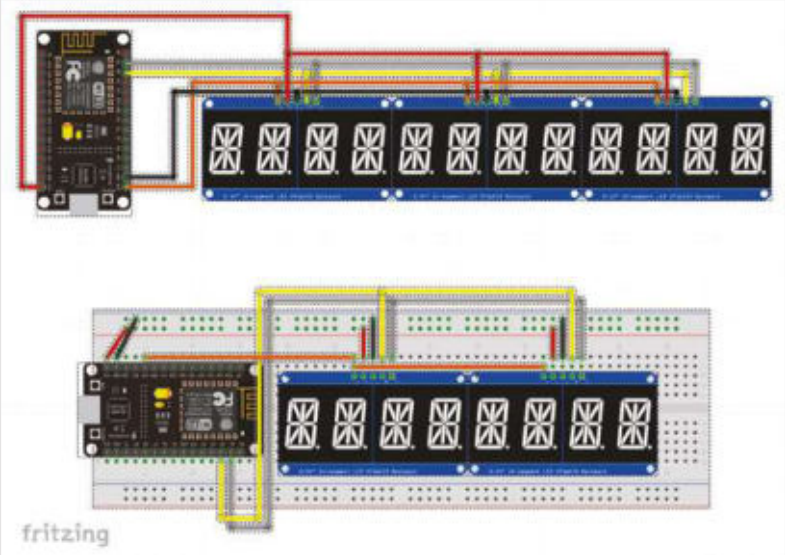
Figure 3
Testing each display not only verifies that it works, but gives you a chance to confirm which one has which address

CONTROLLER CONUNDRUM

You can decide what microcontroller to use for this project (**Figure 4**, overleaf). If you want your mirror to randomly display affirmations from a set list, you can use something simple and offline like an Adafruit Trinket. If you want to try out a more advanced version that connects to the internet to retrieve new messages, you can use a WiFi board like the NodeMCU. Just look up which pins are used for I²C data and clock on your microcontroller and connect accordingly. Additionally, these displays require connections to ground, power, and a reference voltage matching your microcontroller's logic voltage.

YOU'LL NEED

- **Shadow box frame** (Amazon #B07BDR94VW)
- **See-through mirror glass or acrylic** (Amazon # B07TWYKGNB)
- **Black paper tape**
- **Pen or pencil**
- **Ruler or tape measure**
- **Craft knife**
- **USB cable**
- **Adafruit Trinket M0** (Adafruit #3500) **or NodeMCU microcontroller** (Amazon # B07L8W9SP3)
- **White alphanumeric LED displays with I²C backpacks** (two or three, depending on your size preference and budget) (Adafruit #2157)
- **Solderless breadboard**
- **Soldering tools and supplies**
- **Large solderable breadboard** (Amazon #B07DH7QDWT)
- **Solid-core hook-up wire**



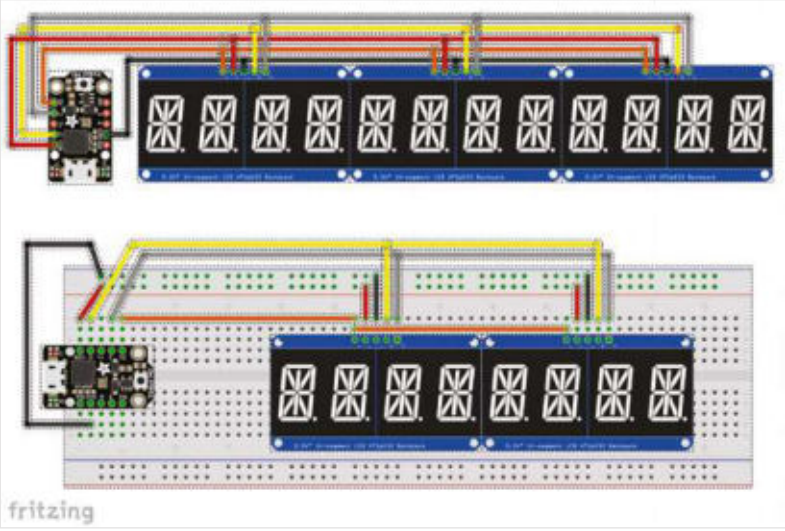


Figure 4
Circuit diagrams for use with Trinket M0 and NodeMCU microcontrollers

with the help of a ruler (**Figure 5**). Note which holes in the perfboard align with the display headers, then solder the display headers in place. It's very important that the displays align with one another such that they will appear as one long seamless display. However, it's not essential that you solder them in the same order you had them in earlier, since you can always rearrange the addresses in the code.

Solder up the rest of the circuit using solid-core hook-up wire, and make sure it works as expected (**Figure 6**). You can download the code from **hsmag.cc/issue30**. Adjust the display I²C addresses if necessary, as you did when testing, so the message scrolls smoothly across all the displays as one.

To assemble the mirror, grab a piece of see-through mirror plastic or glass that is the same size to fit your shadow box frame. You can order this stuff online or find it at speciality plastics and glass retailers. You can

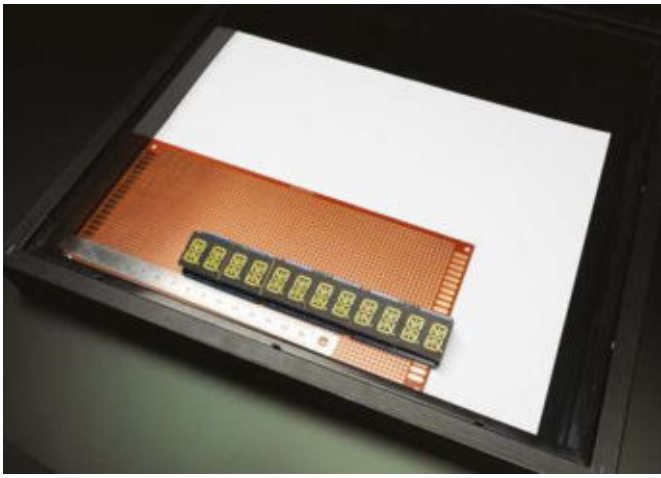


Figure 5
The bright LEDs will shine through the mirror at the lower edge, where the perfboard rests inside the shadow box frame

use a thick piece of glass or plastic instead of your frame's glass, or use a thin piece of plastic layered behind the frame's original glass. Peel away any protective film and assess which side gives the best reflection, since it's not always the same from both

> It's very important that the displays align with one another **such that they will appear as one long seamless display**

sides. Cover the 'wrong' side of it in black paper tape, or tape some on some dark paper.

Place your frame face-down on your work surface, then install the front glass/plastic panel, with the reflective side facing down. Place your completed circuit in the frame, in the same corner it was when you aligned your displays earlier, and use a pencil or pen to trace (as best you can) the outline where the

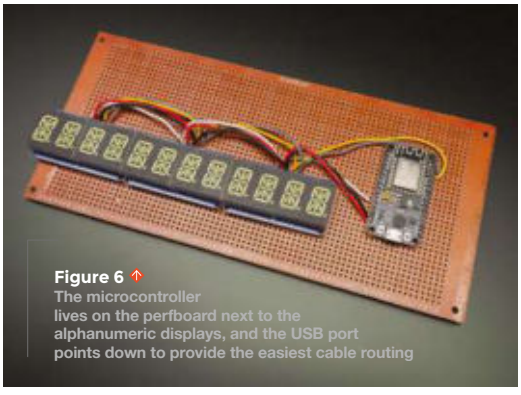


Figure 6
The microcontroller lives on the perfboard next to the alphanumeric displays, and the USB port points down to provide the easiest cable routing



Figure 7
Black paper tape (affectionately abbreviated 'BPT' in the film industry) is preferred for this project because it blocks light out completely

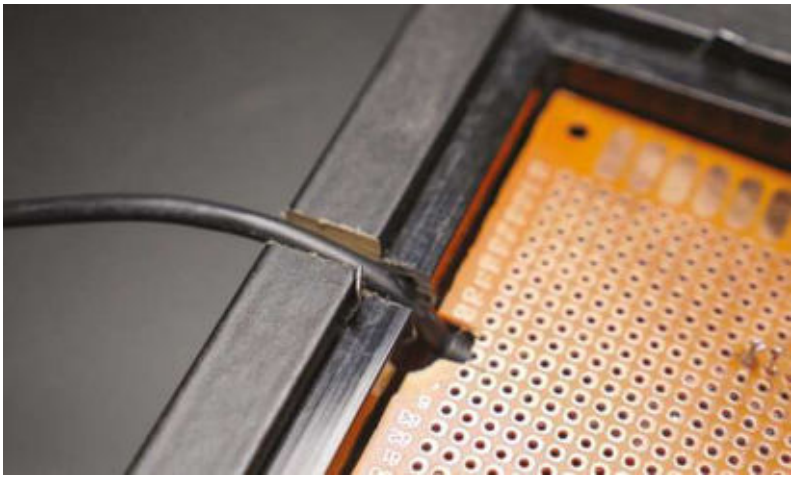


Figure 8
Cable routing: every power cable needs an escape path!

display meets the mirror surface. Remove the circuit and clean up your tracing with a ruler to create a more precise rectangle that matches the size and position of the alphanumeric displays.

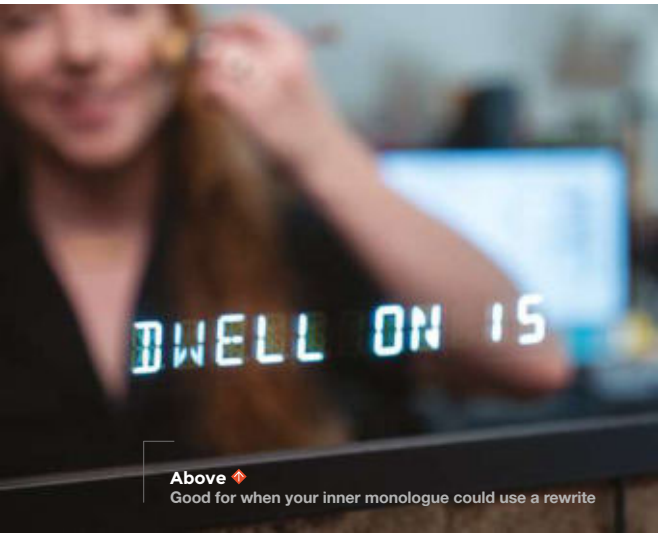
Carefully cut out this rectangle with a craft knife, optionally re-taping the edges of the rectangle to create clean, straight lines (**Figure 7**). This tape will stop light from bouncing around behind the mirror, which would reduce its reflectiveness. Now light can only shine through where the displays are.

Plug a USB cable into your circuit before replacing it inside the frame, aligning the display with the rectangular tape cut-out. Notice how your USB cable might need to bend sharply to exit the back or bottom of the frame. Align, mark, and cut a notch in the frame's back panel to accommodate the cable's exit.

MIRROR, MIRROR (ON THE WALL)

If you want to hang your mirror on a wall, you'll also want to cut a notch in the frame to allow the USB cable to pass through while keeping the frame flush to the wall (**Figure 8**). Depending on your frame material, this can be achieved with a utility blade or small handsaw.

Once you've determined your cable-routing strategy, you can close up the frame. Use more tape to secure your circuit board inside the frame, so it doesn't slide around. The sandwiching force of the shadow box back panel may be enough to hold your circuit in place, but the tape helps you align it perfectly and provides extra insurance it won't wiggle out of place later.



Above
Good for when your inner monologue could use a rewrite





Experimental Cura supports

Pick the right support type for your 3D printing



Ben Everard

@ben_everard

Ben is a maker and keen cyclist, which means he has a keen interest in the air around him and the skills to find out what's in it.

Above Conical supports are much like traditional supports, but with less volume

When you're 3D printing, you build up an object layer by layer – but what if you need to add some plastic where there's nothing on the layer below? Sometimes you can get away with this if it's a short gap (or bridge), but other times, it's just not possible. In these situations, you need supports. These are bits added to the model to support it during printing and can later be snapped off. It sounds so simple, but they can be difficult to remove, and they slow down printing.

Standard supports are printed under the area that needs supporting, and go up until they reach the model. However, there are two more sophisticated options in the experimental section of Cura:

Conical supports The principle behind this type of support is that if the support tapers out towards the top (like a cone shape), you don't need to use as much support material.

Tree supports These start off as a trunk, but can branch out towards the top. Similar to conical supports in that they can support more area at the top than they take up on the build plate, but this can be multiple different supports at the top from one point on the base.

THE TEST

We selected a print of a geometric panther (by Nbarb01 on Thingiverse – **hsmag.cc/mzRVHy**) as this needed quite a lot of supports to print properly. We also enabled the fuzzy skin experimental option because we were feeling particularly experimental. All three support types printed properly – see the box (opposite) on how to enable them.

Traditional supports These behaved exactly as we would expect supports to. This setting used 6.01 m of filament at 10% support density.

Conical supports This used 5.99 m of filament, yet still printed as well as with full supports. While this is quite a small difference, the model had short supports – taller objects would have a bigger saving. Because the supports are thinner at the bottom than the top, it was easier to remove them because they were separate from other parts of the model (in this case, the legs).





Tree supports This used 6.05m of filament, so slightly more than traditional support – however, this is partly due to the particular model being used. The supports themselves were much tougher than either the conical or default supports. They also detached from the model easily. However, since the support was sturdy, it was difficult to remove this from the intertwined legs without damaging them. Again, this is a property of the particular model.

WHICH TO USE?

Both conical and tree supports are classified as experimental, so there's a chance that you could get strange results from time to time. However, if you're willing to take the risk of the odd model going wrong, there's a lot to like about these support options. Conical supports seem like a drop-in replacement for traditional supports, and should work particularly well when there's a large area that needs supporting.

Tree supports are a very different beast. In some cases, they could offer a significant time and material saving, but in others, this won't be the case. They're best suited to models where there are a lot of little areas that need support. They can also branch into a model from outside, so you can have supports off the build plate in cases where this isn't otherwise possible, and this means that less of your model will be marked by support removal.

Above
Tree supports flare at the top, and attach in a different way that makes them easier to remove

Left
Tree supports branch out from the base, but aren't as efficient as conical supports for larger areas

ENABLING **THESE SUPPORTS**

The type of supports you can use will depend on the slicer you use. At the time of writing, Cura had the most flexible support options, so we tested with this.

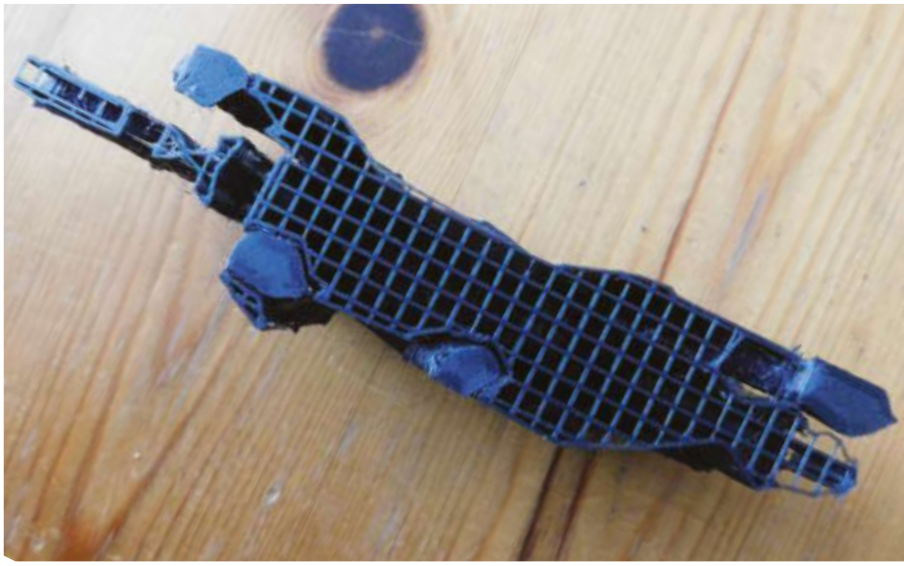
Before you can use conical or tree supports, you'll need to enable them in the Experimental section of Cura.

Click on the cog next to any of the options in the Print Settings box, and then in the pop-up window, you can scroll down to Experimental, and enable either Tree Supports or Conical Supports. This will add extra options to the Print Settings box where you can select these type of supports.

The two support types seem to be implemented in different ways. When using conical supports, you need to enable the Generate Supports option and check the Conical Support box. With tree supports, you only check the Conical Supports box and not the Generate Supports.

With both options, there's a range of variables you can tweak to optimise the supports for your model.

Below
These supports will be familiar to most people with 3D printers. They use a lot of material and can be difficult to remove



DON'T MISS THE **BRAND NEW** ISSUE!

SUBSCRIBE FROM JUST £5

- **FREE!** 3 issues for the price of one
- **FREE!** Delivery to your door
- **NO OBLIGATION!** Leave any time

FREE PI ZERO W STARTER KIT*

With your 12-month subscription to the print magazine

magpi.cc/12months

* While stocks last

Buy online: store.rpipress.cc

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

PG 116

DIRECT FROM SHENZHEN:
USB POWER METER

Keep track of how many electrons your project is stealing from your PC

PG 118



BEST OF BREED

The finest budget oscilloscopes around for makers

PG 124



CAN I HACK A KINECT?

Turn a games controller into almost anything



REVIEWS

126 **Adafruit CLUE**



Lots of sensors on a tiny board

128 **Kennedy workbench**



Carry your workshop with you

USB tester

Keep an eye on your USB current draw

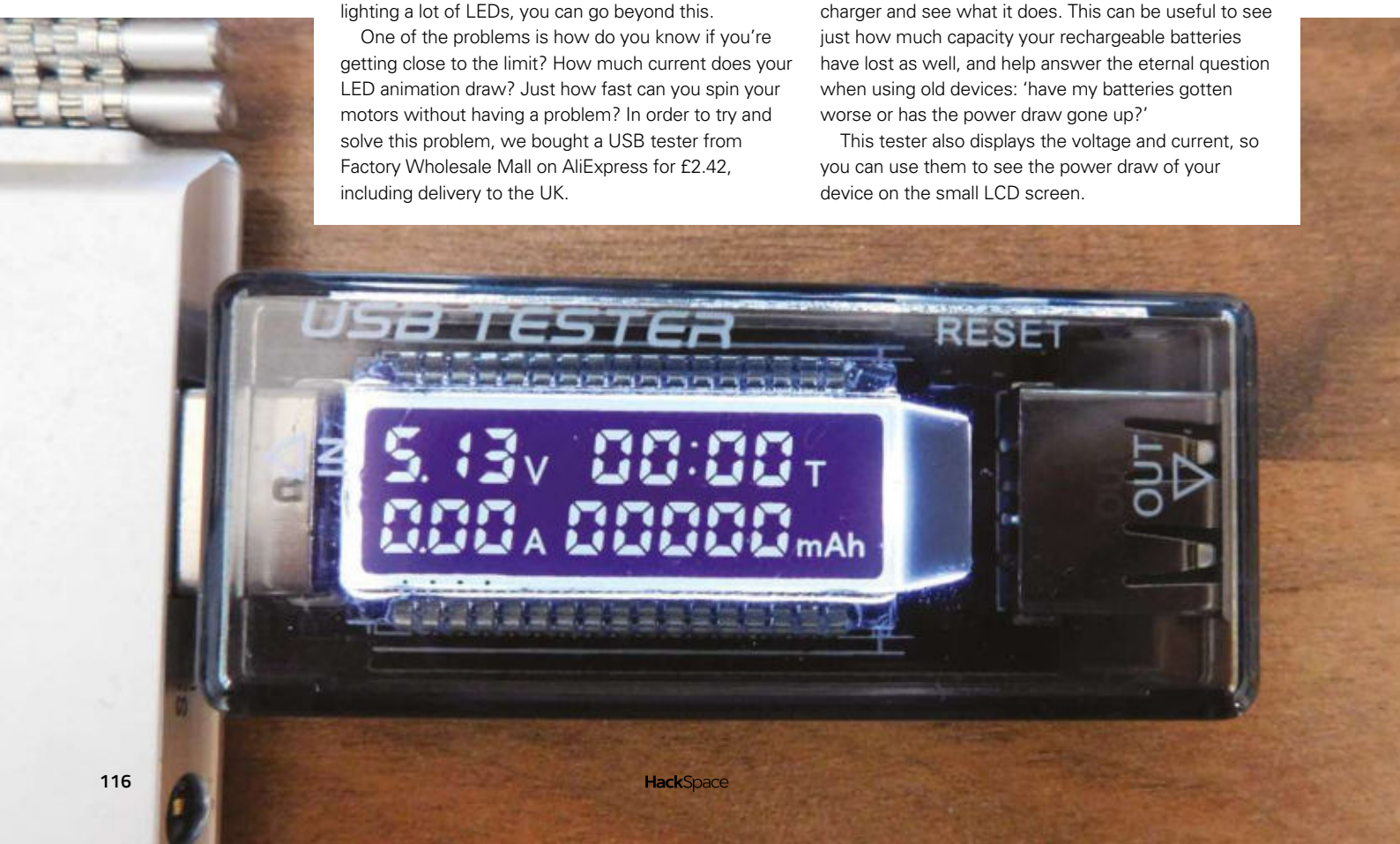
By **Ben Everard** @ben_everard

Like many people, most of our projects start life on USB-powered microcontrollers. The USB connection provides power and also lets us upload code (and read diagnostics). However, there is one problem with this power supply – it can't handle very much current. Exactly how much depends on a few things, but anywhere from 0.5A to 2A is common. If you're powering motors or lighting a lot of LEDs, you can go beyond this.

One of the problems is how do you know if you're getting close to the limit? How much current does your LED animation draw? Just how fast can you spin your motors without having a problem? In order to try and solve this problem, we bought a USB tester from Factory Wholesale Mall on AliExpress for £2.42, including delivery to the UK.

These testers sit in-line in USB connections, so they have a socket on one side and a plug on the other. They come in both standard USB (Type A) and USB-C variants, but we have the standard USB version. Their main purpose is checking the capacity of batteries. For things like phones, you can see how much charge goes in, and for USB battery packs, you can see how much charge comes out. There's a handy counter that adds up the mAh, so you just need to plug it in via the charger and see what it does. This can be useful to see just how much capacity your rechargeable batteries have lost as well, and help answer the eternal question when using old devices: 'have my batteries gotten worse or has the power draw gone up?'

This tester also displays the voltage and current, so you can use them to see the power draw of your device on the small LCD screen.



Below
Add to your USB port for an instant read-out of voltage and current

MICROCONTROLLER TESTING

Data passes through the tester, so you can simply pop it into your computer's USB port and work as you would if it wasn't there. The current draw is there if you ever need to see it, and should it get too high, you can pull it out without waiting for something drastic to happen before you realise that you've exceeded the limits. This author once destroyed a laptop's motherboard by exceeding the USB port's current limit, so you should pay attention to this if there's a risk you might be getting close to the maximum.

It outputs both voltage and current to two decimal places, but the current doesn't appear to be accurate below about 0.1 A. For checking that you're not exceeding current limits, this isn't likely to be a problem, but if you're going the other way around and trying to reduce your power consumption as far as possible, you'll probably need a different solution.

We've been using this for a little while now, and it's really useful, particularly with medium to large NeoPixel projects. With these, it can be really hard to know how much power you're drawing. By testing with this, and using low brightnesses, you can actually do a lot with USB power.

It can go the other way as well, and you can check that your power supplies can really deliver the current they promise, particularly with higher draw devices such as Raspberry Pi boards. In this case, you'll probably notice the voltage drop as the current increases. If it drops too far, then you'll no longer be able to use the device. With this device, you can see exactly what current draw that starts to happen at, and see whether the problem is that you're using more power than you thought or if your power supply can't deliver as much as it should be able to.



The usefulness of this device is in its simplicity. There's nothing in this tester that you can't do with even the most basic multimeter, but you can do it without needing to do anything more than plug in a USB cable. Inserting a multimeter into your power system can be a bit of a challenge, especially if you're using expansion boards rather than wiring it up completely yourself. For beginners, you don't have to worry about getting the multimeter in the right place, and even advanced users will appreciate the simplicity and tidiness of this solution.

For less than £3, you can use this and just not have to worry about accidentally exceeding current limits. Oh, and if you're ever suspicious that a battery might not have its stated capacity, you can check that as well.



Above
The tester is slightly fatter than our laptop, but just about fits in the USB port when it's flat on the desk

Below
It's small enough to fit into any electronics kit

DIRECT FROM **SHENZHEN**

ONLY THE BEST

Your first scope

Can you really buy an oscilloscope for under $99? Yes!

By **Marc de Vinck** @devinck

When you first start out learning about electronics, you most likely don't need an oscilloscope. In fact, you most likely don't even know what one is when you first start. And when you do learn about what they are, and what they do, you also learn that they are expensive. Well, the truth is, you're right: you don't need one when you first start out. But you're also wrong, because you can get a decent beginner-friendly oscilloscope for a reasonable price. And we know all those seasoned electronics enthusiasts are going to argue that a decent oscilloscope costs at least $300, and very quickly go up to several thousands of dollars, and they are right, too. But if you are just starting out with basic circuits, playing around with an Arduino, or Raspberry Pi, a basic oscilloscope can cost much less and still prove to be helpful.

So what is an oscilloscope? In very simplistic terms, it's a device that is used to measure changes in electrical signals over time. They can do this at a very fast rate, and the more you pay for one, the faster and more accurate they become. They can measure frequency, time, phase shifts, and more. For a beginner, what you typically want to see is voltage over time, and typically not at a fast rate. And fast is a relative term!

Picking the right oscilloscope is a huge topic that typically leads to a lively debate! You have to consider price, functionality, accessories, accuracy, and dozens of other characteristics. But in this Best of Breed, we'll be looking at some simple, relatively slow, beginner-friendly oscilloscopes that cost less than $99.

> For a beginner, what you typically want to see is **voltage over time, and typically not at a fast rate**

These oscilloscopes are the perfect tools to watch your Arduino or Raspberry Pi pins go from high to low, or trouble-shoot a simple analogue or digital circuit. It's almost magical the first time you see your pin trigger the scope signal on and off at an unimaginably fast rate. And the more you learn about electronics, the more you will want a better, more powerful oscilloscope. But these oscilloscopes are a great start. And when you do eventually upgrade, which you will, maybe you can pass down your first-time scope to someone who is just getting started on their own journey into electronics.

DSO Nano v3 vs WAVE2 2-Channel Oscilloscope

SEEEDSTUDIO ◈ $89 | seeedstudio.com

JYE TECH ◈ $93 assembled ($79 for the kit version) | accudiy.com

The DSO Nano v3 is the latest variant of the pocket-size compatible 32-bit digital storage oscilloscope line sold by Seeed Studio. It's built around the ARM M3 coupled to a 320×240 colour display and an SD card. It's also really compact, and it features a metal case for added durability. And although it's not as powerful as most benchtop oscilloscopes, the portability makes it a really valuable tool for beginners and experts alike.

The Nano v3 version comes with new software that's more user-friendly, and upgraded firmware. It also features waveform storage and playback, six triggering modes, measurement markers and signal characteristics, and a built-in signal generator. This author has personally owned a couple of the older versions of the DSO Nano, and they come in really handy – more than you would think. I love being able to bring mine right to where I'm working, even if it's only a few feet away, rather than try to move the sometimes fragile circuits I'm prototyping, or lug around my benchtop model.



Left
A compact scope for measurements on the go



Above
The minimal device hides some powerful features

The WAVE2 is a two-channel, 200kHz per channel, oscilloscope that features an easy-to-use colour touchscreen display, allowing access to many features only found on more expensive oscilloscopes. In addition to being a compact and capable two-channel little scope, it features a built-in function generator and X-Y plotting. Those are features usually reserved for more expensive oscilloscopes. You can purchase the WAVE2 as a fully assembled product, or as a DIY kit. Both come with everything you need, including a USB cable and two high-quality BNC probes.

> "You can purchase the WAVE2 as a **fully assembled product, or as a DIY kit**"

VERDICT

DSO Nano v3

Compact and rugged.

10/10

WAVE2 2-Channel Oscilloscope

Affordable, with extra features.

8/10

BEST OF BREED

ScanaQuad Logic Analyzer & Pattern Generators

SCANAQUAD LOGIC ◈ **$98** | **pimoroni.com**



A **few of you will instantly know the ScanaQuad Logic Analyzer & Pattern Generators isn't an oscilloscope; it's a logic analyser.** We know! But for many people the end results, or desired information to gather, is similar, so that's why we included it in this review. We also included it because it's under $99, which is a great price for a logic analyser.

The main difference between an oscilloscope and a logic analyser is that, typically, an oscilloscope is measuring analogue voltage over time, and a logic analyser is measuring the digital equivalent at 1-bit resolution. The other big difference is a logic analyser will typically have many more channels to listen in on. Ultimately, you will most likely want both types of tools on your bench, but most start with a standard oscilloscope for its versatility.

When you are ready to start looking into the depths of logic level signals, you'll find that the ScanaQuad Logic Analyzer, combined with the ScanaStudio, is perfect for capturing and analysing digital signals easily and affordably. It's really handy when working with embedded systems, as long as you have a computer nearby to run the software.

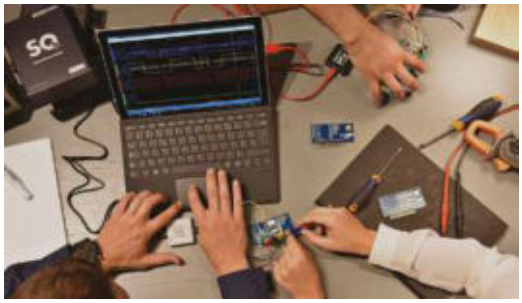
There are a lot of features, including packet viewing, search and filtering of captured data, unlimited annotations, markers and measurement tools, and the ability to export your data. You really need to head over to the website to read more about this analyser, and you can also download the software and take it for a spin to see if it will meet your needs.

RIGOL DS1054Z

RIGOL ◈ **$449** | **adafruit.com**

The Rigol DS1054Z from Adafruit certainly doesn't make the list of oscilloscopes for under $99, but we wanted to include a great choice for your next oscilloscope when you need more functionality. And although this four-channel, 50MHz version sells for around $500, it's still a lot less expensive than a more robust and faster, professional-level scope which can cost thousands of dollars or more. Check out the product page for more information about this advanced oscilloscope. It's a good one!





Above
Peek into the hidden world of digital signals with a logic analyzer

Below
The PC software unlocks the full features

VERDICT

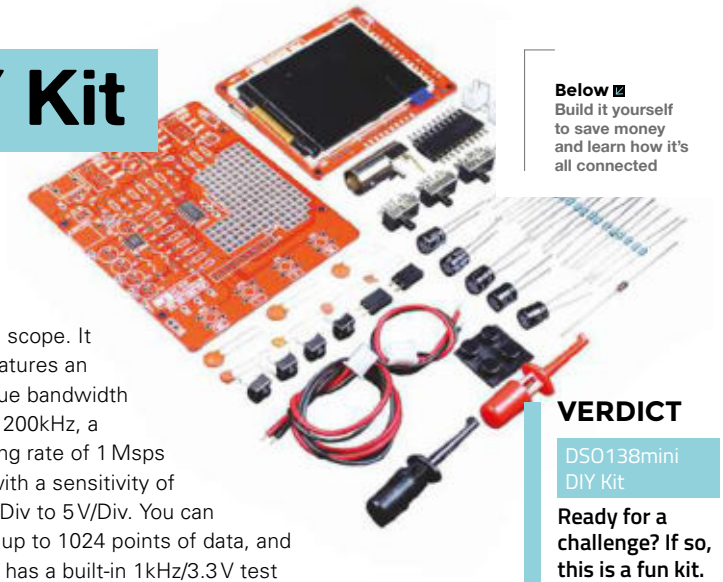
ScanaQuad Logic Analyzer & Pattern Generators

A good tool to consider after you have a scope.

8/10

DSO138mini DIY Kit

JYE TECH ◆ **$32** | jyetech.com



Below
Build it yourself to save money and learn how it's all connected

So, you are well on your way to wanting to learn more about electronics, and you consider yourself to have decent soldering skills. Well, do I have a kit for you! DSO138mini DIY Kit is a fairly capable oscilloscope, once you solder it together. And although that will take some time, it's fun, and not only do you save some money, but you also get the satisfaction of building it yourself.

The DSO138mini features an RGB LCD with a resolution of 320×240, which is fairly good for such a small scope. It also features an analogue bandwidth of 0 to 200kHz, a sampling rate of 1 Msps max, with a sensitivity of 10 mV/Div to 5 V/Div. You can record up to 1024 points of data, and it even has a built-in 1kHz/3.3 V test signal. Once assembled, you'll have yourself a very capable little scope. And it's always fun to build your own tools!

VERDICT

DSO138mini DIY Kit

Ready for a challenge? If so, this is a fun kit.

9/10

Pocket Science Lab Dev Board

POCKET SCIENCE LAB ◆ **$59** | us.pslab.io



Left
Why have a built in screen when we've all got smarphones and PCs?

The Pocket Science Lab Dev Board is a fairly small electronics accessory for your Android phone or PC that does a lot. It's more than just a four-channel up to 2 Msps oscilloscope. It has a built-in 12-bit voltmeter, 4MHz logic analyser, two sine/triangular wave generators, and a power source. And since it's open source, you never know what others might add in functionality. At this price, it's worth it for just the oscilloscope features. All the other features are just a great bonus. Be sure to check out the website for more information, because there are a lot!



VERDICT

Pocket Science Lab Dev Board

A great accessory for Android and PC users.

10/10

BEST OF BREED

Love To Code Chibi Scope

CHIBITRONICS $30 | adafruit.com

Just like it says on the Adafruit website, "What is this adorable little data dumpling?" Good question! The Chibi Scope is a fun and affordable oscilloscope that works in conjunction with the separately sold Chibi Chip and Chibi Clip. That may sound fairly limiting, but it's not. The Chibi Scope works when communicating at 9600 baud, so it will also work with other development boards too, such as the Arduino or Raspberry Pi.

> "**It's fun, it's small, and hopefully it's inspiring for** getting a younger audience excited about learning electronics"

The Chibi Scope also has some other built-in features, like a simple voltmeter and ASCII text display mode. So you might be wondering how well the oscilloscope works for measuring the voltage over time. The simple answer is, for the price, it works fine. But, don't think this is going to give you fast sample rates or multiple channels. Because it won't. Where the Chibi scope really shines is being able to show when a pin goes high or low, or how many volts are flowing when a sensor is triggered. It's fun, it's small, and hopefully, it's inspiring for getting a younger audience excited about learning electronics.

Below
Not everyone needs a complex scope and sometimes simpler is better



VERDICT

Love To Code Chibi Scope

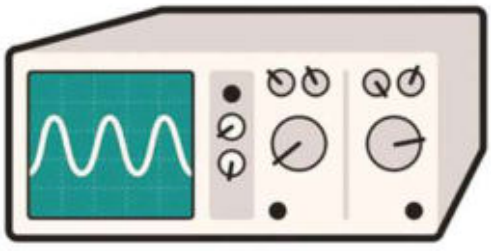
Good for a younger enthusiast.

8/10

OSCILLOSCOPE – STICKER

ADAFRUIT $1.50 | adafruit.com

OK, this is the least expensive oscilloscope in this round up! Yes, it's just a sticker, but it's also awesome. Adafruit has a lot of cool electronics-themed stickers that would make a thoughtful, inexpensive, and fun gift to give your favourite electronics enthusiast. Just peel off the back and slap it on your computer, workbench, or notebook. Be sure to check out all the different electronics-related stickers on Adafruit's website.



GET STARTED WITH ARDUINO®

Robots, musical instruments, smart displays and more

£10 with FREE worldwide shipping



Inside:

- Build a four-legged walking robot
- Create a Tetris-inspired clock
- Grow veg with hydroponics
- And much more!

AVAILABLE NOW hsmag.cc/store

plus all good newsagents and:

WHSmith BARNES&NOBLE

Available on the App Store

GET IT ON Google Play

FROM THE MAKERS OF HackSpace MAGAZINE

Can I Hack It?

An Xbox 360 Kinect?

Can we reuse an old Xbox 360 Kinect?



Les Pounder

@biglesp

Les Pounder loves taking things to pieces and seeing how they work. He teaches others how to be makers and tinkerers at events across the UK. He blogs at **bigl.es**

In the last issue, we tore apart an old Xbox 360 controller, so this month we continue looking at old console tech with the Xbox 360 Kinect. Released back in 2010, the Kinect for Xbox 360 was a bold attempt at using cameras as input devices. Featuring a 640×480 camera, and a 3D scanner called Light Coding, which pieces together 3D scenes using images and data from an infrared sensor. The unit also has a microphone used for voice interaction and for noise cancellation in group chats.

With all this tech, the Kinect must have been a hit? Well, no, it took some time and a few revisions, but the project flopped, and it is no longer being used for gaming.

But what can we do with this piece of kit? Let's take one apart, and see what makes it tick.

GENERAL CONSTRUCTION

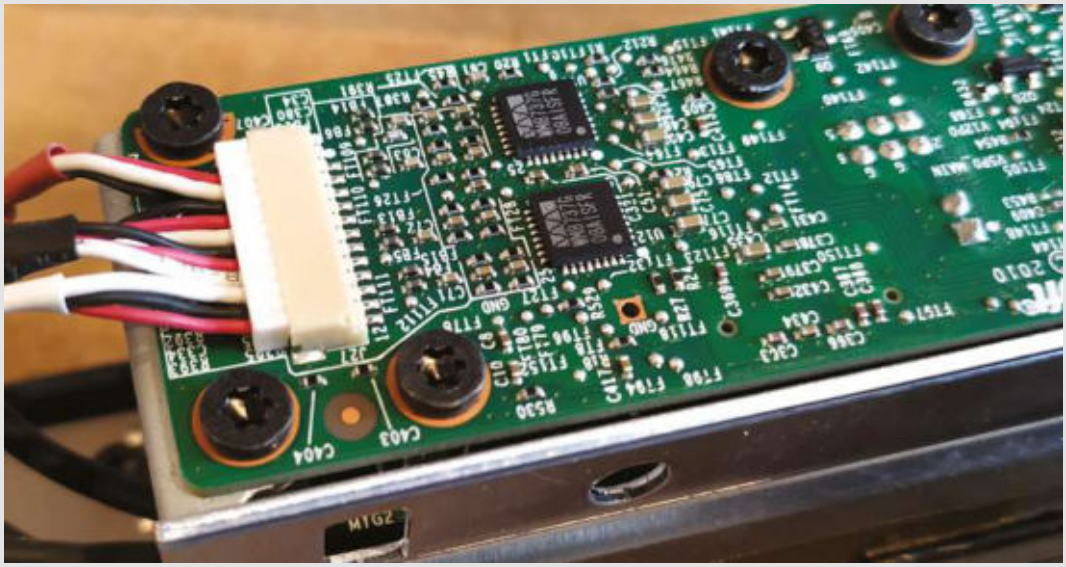
This is a solidly put-together piece of kit which is made from overlaid sections of plastic that interlock with tabs and spacers. To take it apart you will need some spudgers and a T7 and T8 security bit, which feature a hole in the centre of the bit. The unit is made of two main sections. The base contains a small, but powerful, motor which can be accessed by removing four T7 screws, hidden under the rubber base. The camera unit has two T8 screws hidden under XBOX 360 stickers. Then there are a series of four T8 screws hidden under a metal grill on the underside of the camera unit. The metal grill is held using a mix of tabs and adhesive. The metal will deform a little when spudgered, but it will bend back into shape.

ELECTRONICS

Inside the camera unit is a densely packed series of boards connected electronically using a series of JST connectors and thin flat flex/ribbon cables. The boards are physically connected using 5 mm standoffs and held in place with T8 screws. The modular nature of the electronics means that we can take the unit apart with confidence that we can put it back together again. The small DC fan is next to a flat piece of aluminium, used as a simple heatsink for the processor. All of the cameras can be removed by unlocking the plastic catches on the connectors and then carefully removing the flat flex.



Above
A little beat up, but still perfectly functional. The Kinect offers a fun – if a little tricky – sensor for hacks



Left
The circuit board is densely packed with surface mount components, but well-labelled and easily accessed with a steady hand

The electronics inside the Kinect are a mixture of custom boards for general CPU, image duties (Marvell) which are closed-source designs with no data sheets, and a stereo mic input (Wolfson Microelectronics WM8737). We also found an Allegro A3906 chip used to control the DC motor in the base used for tilting the camera.

> The Kinect is a bit sparse, so only salvage **if it truly is broken or you explicitly need the bits**

An interesting discovery was the non-standard 'USB' connector which connects to a 12V power supply and your computer. Looking at the male connector face on, with the diagonal cut out to the left: top left and bottom left are 12V, there are three GND pins, and in the bottom right is 5V connected to the computer. The lower centre two pins are standard USB data pins.

HACKABILITY

As a source of parts, the Kinect is a bit sparse, so only salvage if it truly is broken or you explicitly need the bits. For hardware hacks directly to the electronics, the Kinect offers a limited number of options, but the PCB has a good silkscreen to identify the components and their use. The true hackability lies with using the unit as is, but with libraries such as OpenKinect/libfreenect (**github.com/OpenKinect/libfreenect**) and pykinect2 (**github.com/Kinect/PyKinect2**). Installation of OpenKinect requires us to download and build the software manually, and so a computer with a decent CPU is required; Raspberry Pi 4 would be the absolute minimum we would suggest. Both of these libraries grant us low-level access to the technology, and we can use the cameras to detect skeletal structures of up to six persons, measure depth using infrared light, and create 3D maps of rooms using the same light-mapping technology. The libraries helpfully abstract these low-level functions into easily used commands and scripts.

CONCLUSION

The Kinect is a worthwhile item to hack. It may not be the most interesting of items to tear-down, but we learnt that it is a modular unit, which could be re-used inside another shell, and we can hack the software to control it and integrate it into a project. In fact, we have seen one being used to assess how a person sleeps in a bed, to determine the correct mattress for them. Very useful, and nothing to do with video gaming.

Happy Hacking!

OPENKINECT

We spent a day using the OpenKinect software (libfreenect) and were impressed with how easy it was to use. But the stumbling block was the installation process which is not friendly, so take your time and follow it step by step.

We managed to capture live depth data, which could be used in a project as a form of input, for example, a wall of NeoPixels which duplicate the depth data using colour. We also managed to map our room in 3D with surprising accuracy, and produce a 3D model which we were able to fly around in real time!

This defunct piece of technology has a new lease of life, thanks to open-source software and eager makers looking to hack it into their next project.

YOU'LL NEED
Xbox 360
Kinect v1

COST
£ Various

WHERE
Available via eBay, Amazon, thrift stores

REVIEW

Adafruit CLUE

A lot of features in a familiar shape

ADAFRUIT $39.95 | adafruit.com

By **Ben Everard** @ben_everard

This little board will look familiar to a lot of people. It's based on the form factor of the micro:bit that's been popular with schools in the UK and around the world, but comes with a lot more on board.

We can't really skirt around the fact that this is an expensive microcontroller – it's $39.95, not including delivery. However, for that, you get a whole lot of bits and pieces. It's powered by an nRF52840 microcontroller which has a 64MHz ARM Cortex-M4 processor and 256kB RAM for crunching data, and Bluetooth Low Energy (BLE) for communication. On the front, the noticeable features are two buttons and a 240×240 pixel screen.

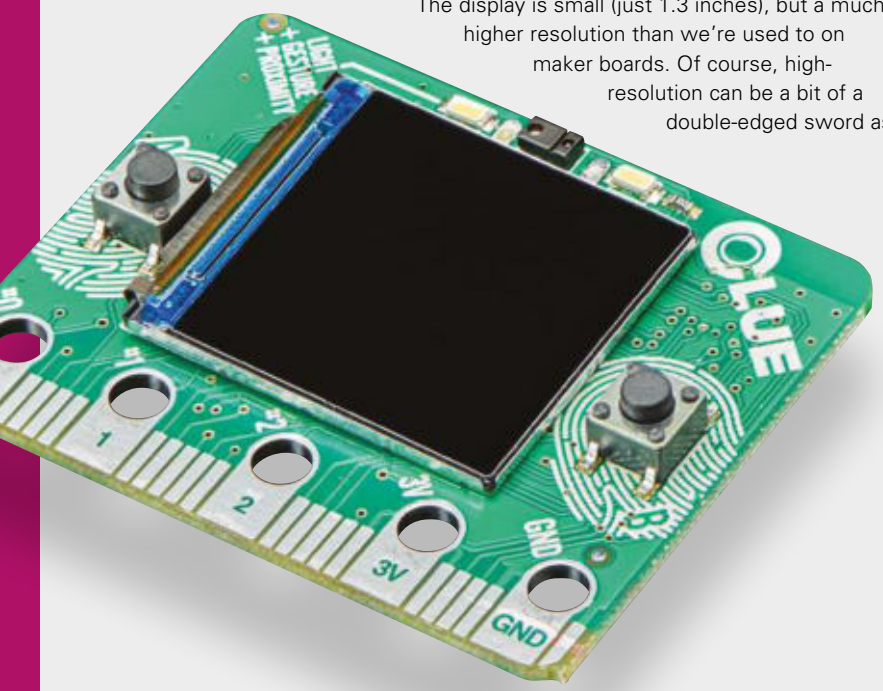
The display is small (just 1.3 inches), but a much higher resolution than we're used to on maker boards. Of course, high-resolution can be a bit of a double-edged sword as it requires more processor power and RAM to be able to make use of it. Here again, the CLUE is well provisioned. This, combined with the recent speed improvements in the CircuitPython Displayio module, makes this display easy to use. It can also be used with the Arduino IDE.

Less noticeable is a whole bunch of other features. The full list is a little intimidating:

- STMicroelectronics series 9-DoF motion – LSM6DS33 accelerometer/gyroscope
- LIS3MDL magnetometer
- APDS-9960 Proximity, Light, RGB, and Gesture Sensor
- PDM microphone sound sensor
- SHT Humidity
- BMP280 temperature and barometric pressure/altitude
- RGB NeoPixel indicator LED
- 2MB internal flash storage for data logging, images, fonts, or CircuitPython code
- Speaker for playing tones and beeps
- Two bright white LEDs in front for illumination or colour sensing

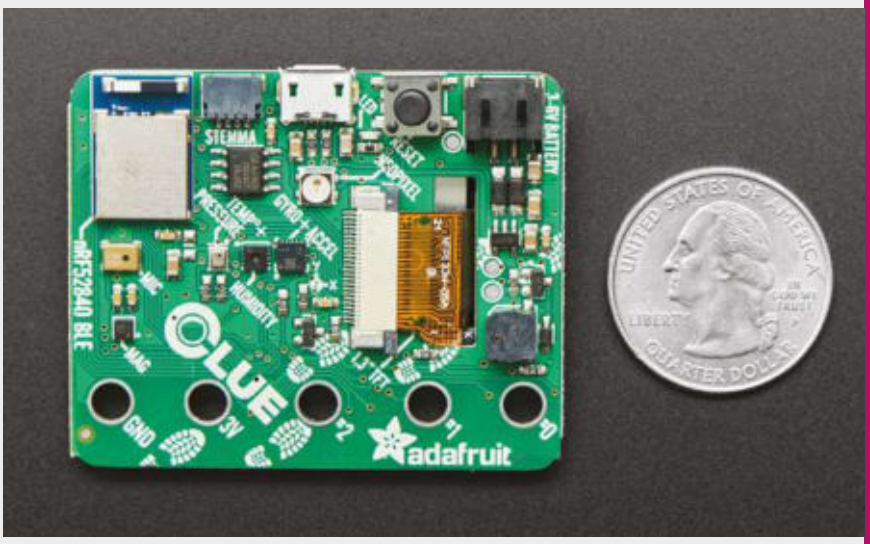
It's impossible to go through all of these in detail in a two-page review. We've gone through a few of the features in more detail in a tutorial on using CircuitPython on page 78. All the sensors are reliable, but maker level. For example, the microphone can record samples at 16000Hz, which is fine for a lot of uses, but don't expect high-quality audio out of it.

Below
From the front, the inspiration of the form factor is obvious



SHAPE-SHIFTING

There are good and bad things about the micro:bit form factor. It's a good size for handheld devices (even for little hands), and the buttons either side of a central display are pleasing to use.



On the downside, we're just not fans of the bottom connector on the micro:bit. It can be used with crocodile clips, but only three IO pins (plus 3V and GND) can be accessed this way, and even then, only if you're very careful to avoid touching any of the smaller pins. Banana plugs connect well to these holes, but not many other things connect to banana plugs. You can get banana-to-crocodile leads, which work great for the micro:bit, but only the micro:bit. If you're planning on using a lot of external hardware with the CLUE, it's worth considering a breakout adaptor such as Kitronik's, which will make the IOs breadboard compatible.

The saving grace of the bottom connector is that it lets you push the micro:bit into expansion hardware and still access the buttons and display on the board. Note, though, that hardware designed for the micro:bit may or may not work with the CLUE. Software support is a particular problem as there's no Scratch version for the CLUE's microcontroller.

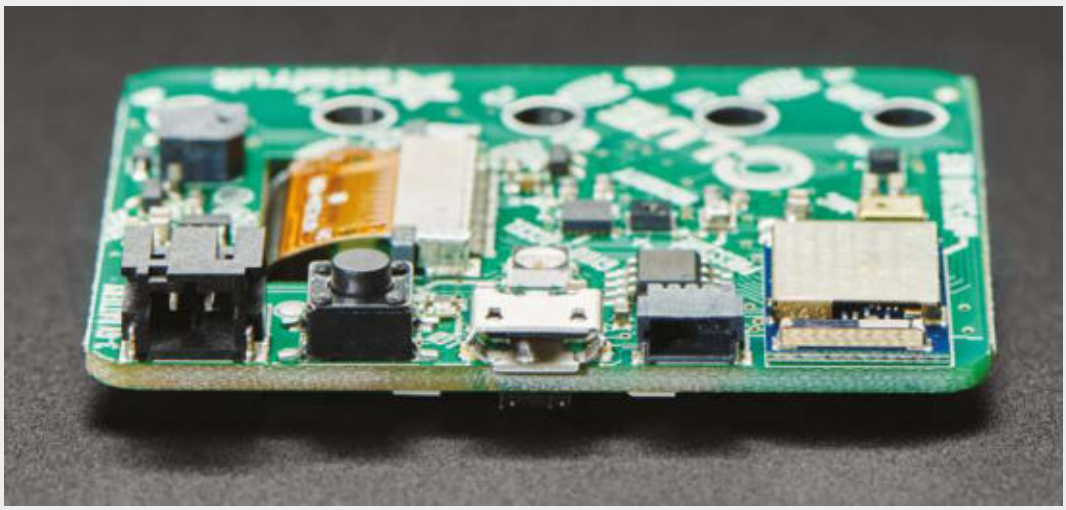
The CLUE extends the standard micro:bit form factor slightly by including a STEMMA QT connector which makes it easy to hook up I²C hardware. There are lots of expansions that are already compatible with this, including SparkFun's Qwiic range and Grove's I²C sensors (though the latter will require an adaptor). This, in our view, is a big advantage because not only is there a lot of hardware available off the shelf for this, but it's easy to build your own.

LOTS OF POSSIBILITIES

That's a lot of bits to play with, and that's a good way of thinking about the board – a tool for learning about how to program a microcontroller with a lot of different hardware. There are probably few projects where it's cost-effective to buy a CLUE rather than a more stripped-down board plus the bits you need. However, if you're interested in learning about embedded computing – and particularly the programming side of it, then the CLUE is a great option because, while it might not be perfect for any one project, it's great for lots and lots of projects.

> If you're interested in learning about embedded computing ... **then the CLUE is a great option**

We really like boards like this for learning and experimenting with. There's so much bundled on the board that you can try out an immense amount without getting caught up in wiring. The inclusion of the STEMMA connection, in our view, is significant because this makes it much easier to hook up additional sensors and actuators. □



Above
Despite all the features, it's a small board

Left
You can power the CLUE via either USB or the battery connector

VERDICT

Plenty to play (and learn) with in one small package.

9/10

REVIEW

Kennedy portable workbench

Get making wherever you are

KENNEDY £24.79 | cromwell.co.uk

By Jo Hinchliffe @concreted0g



Above
The portable Kennedy workbench comes with some reconfigurable blocks that slide into aluminium T-slots, enabling numerous ways of work holding

Below
The upper tray with tools has legs on the back of it to keep it in position when open. Modifying this small vice to fit to the T-slots has made a useful accessory for smaller work

We are big fans of portable tools and equipment that enable makers to carry on working on their project in different environments. From the backs of vans, fields, maker events, and everything in between, carrying tools and a workbench can be a challenge. So, we were keen to review this portable workbench that also offers some tool storage.

The Kennedy portable workbench is available to order online from numerous stockists, but we ordered ours from Cromwell Tools, as it was discounted at the time of purchase to just over £20. It was delivered promptly and came in a simple cardboard cover.



Opening the packaging reveals some instructions, but it's pretty intuitive. The handles to wind the jaws/work areas are stowed in a small box in the upper drawer storage section, and they clip into holes to engage the lead screws to move the table and clamp work. It's nice that these are easily removable so that it's easy to pack and carry without the handles catching on things.

There are some supplied blocks that neatly slide into the aluminium extruded T-slots that allow for items to be clamped in different ways, and these are thoughtfully designed with a slip-on plastic block that enable different geometries of workpieces, like round bars or square stock, to be held in different orientations. Positioning items to be held vertically or horizontally, using either the jaws of the bench table or the plastic blocks was simple, and the clamping force between the jaws feels the same as any other workbench we have used.

Fold-out feet on the base of the unit are nicely designed to be stowed in transit, and they have a hole in each one so that they can be screwed into a surface. The feet are also flat-surfaced, allowing the user to G-clamp the unit down to a table. We also considered that if you were working outdoors and needed extra rigidity, you could peg/stake the



unit into the ground. When folded in, the feet also help retain the lower of the two storage areas. The workbench has numerous useful markings on its work surfaces. While perhaps not useful for super-accurate work, the rulers and angle markings are usable for quick marking out or for checking rough dimensions.

AT CAPACITY

The storage areas of the workbench are made from hard, injection-moulded plastic and seem very rugged. It's pretty intuitive, but it's worth noting that you need to lift/carry the workbench from the handle which is a part of one of the drawers, or there is a chance that the drawers will remove themselves. You also have to rotate out the feet and slightly open the upper drawer to allow the lower tray to be removed from the bench. Having packed the drawers with tools, we found the bench carried them securely with no flex, although with the weight of the bench, you may wish to choose lighter tools to go in it, as it can become very heavy quite quickly.

> You may wish to choose lighter tools to go in it, **as it can become very heavy quite quickly**

We really liked the inclusion of T-slots and have had a few thoughts about how they could be used to extend the versatility of the bench. We plan in the future to 3D-print some extra work-holding blocks, for example. We have already made one little additional device. We had a small, cheap vice that was originally designed to clamp on to a table. We had upgraded to a better vice, so this was a spare that had sat unused for a while. We removed the clamping base from the vice and then drilled two 6 mm diameter holes through the base to allow us to use some 6 mm threaded bar and some square nuts we had that fitted the T-slots. It is great to be able to add a small vice to this bench, and we hope it's the first of many extra uses we find for it. □

Below
The folding feet secure the lower tool tray when in transit, and help secure the bench in place when rotated out

VERDICT

For the price, it is difficult to fault this portable workbench.

10/10



next month

issue #31 ON SALE 21 MAY

BUILDING THE INTERNET OF THINGS



ALSO

- MUSIC
- 3D PRINTING
- RASPBERRY PI
- CIRCUITPYTHON
- AND MUCH MORE

DON'T MISS OUT

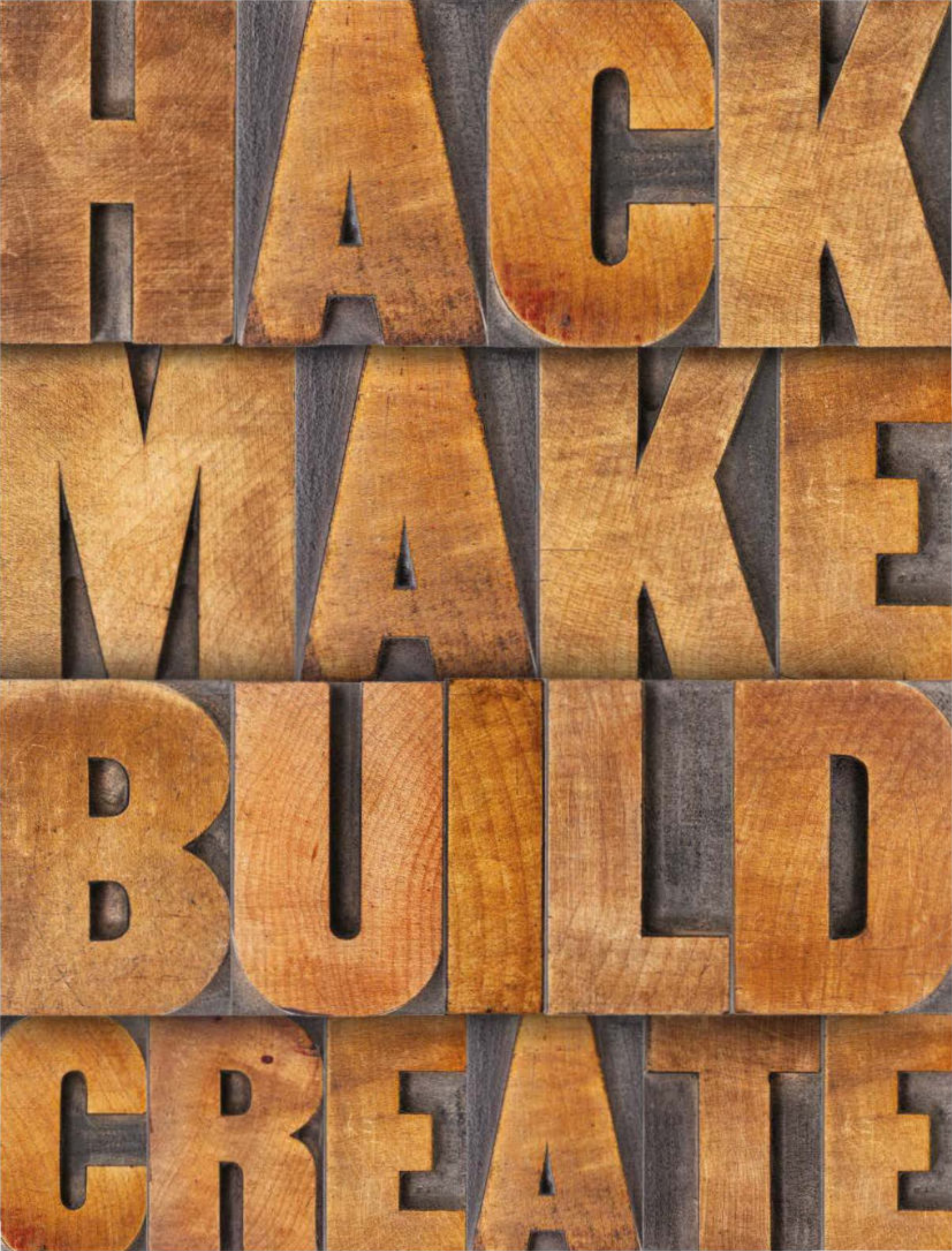
hsmag.cc/subscribe



"I learned everything simply by failing… by crying in the middle of the night, wanting to throw everything away in pure desperation at the thought that I'm too dumb to do it right.

This is really what cosplay is about: it's not about making a cool costume; it's about failing and crying but not giving up – just keeping on doing it, and doing it better."

Kamui Cosplay


HACK
MAKE
BUILD
CREATE